Petitioner RPX's Demonstratives
November 8, 2016 Oral Argument

RPX Corporation, Petitioner

v.

Applications In Internet Time LLC, Patent Owner

IPR2015-01750, Patent 8,484,111

IPR2015-01751, IPR2015-01572, Patent 7,356,482

RPX Exhibit 1063
RPX v. AIT
IPR2015-01751

'482 patent discloses a multilayer application system that detects the changes in the regulatory and non-regulatory laws that arise in various commercial and industrial activities.

III. RPX is a Proxy for Real Party in Interest Salesforce.com, Inc.

In its decision instituting these three trials, the Board stated that there was insufficient evidence to find that the real party in interest is Salesforce.com, Inc. Patent Owner disagrees with the Board's view of the law and the facts, and in particular believes that the Board misconstrued the law. As explained previously, the AIA was intended to prevent defendants from getting "a second bite at the apple." Yet, the Board is doing just that by allowing Petitioner to act indirectly for Salesforce. In its decision, the Board set an improperly high burden of proof for the patent owner, and also improperly shifted the burden of proof to the patent owner. As explained in Patent Owner's Preliminary Response, Salesforce is the real party in interest and Petitioner is acting as its proxy. Because Salesforce is time limited, so is Petitioner and patentability should be confirmed on this basis.

IV. Claim Construction

Claims in an IPR are currently interpreted according to their broadest reasonable interpretation. *In re Cuozzo Speed Technologies, LLC*, 778 F. 3d 1271 (Fed. Cir. 2015). The current standard for claim construction by the Board is the

POR, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751

US006341287B1

(12) **United States Patent**
Sziklai et al.

(10) **Patent No.: US 6,341,287 B1**
(45) **Date of Patent: Jan. 22, 2002**

(54) **INTEGRATED CHANGE MANAGEMENT UNIT**

(75) Inventors: **Anthony T. Sziklai**, Half Moon Bay; **Ashish K. Verma**, Foster City; **Judith E. Popowski**, Half Moon Bay; **Richard Frankland**, San Jose; **Christopher M. Mitchell**, El Granada; **Joseph D. Ferguson**, Santa Clara; **Douglas H. Sturgeon**, San Mateo, all of CA (US)

(73) Assignee: **Alternative Systems, Inc.**, Half Moon Bay, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/215,898**

(22) Filed: **Dec. 18, 1998**

(51) **Int. Cl.**[7] **G06F 17/00**
(52) **U.S. Cl.** **707/102**; 705/8
(58) **Field of Search** 707/102, 200, 707/202, 203, 205; 705/7, 8, 9, 10, 28, 29, 11; 588/1, 16; 700/266

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,803,039 A * 2/1989 Impink, Jr. et al. 376/216
5,185,699 A * 2/1993 Reiner et al. 707/531
5,532,928 A * 7/1996 Stanczyk et al. 705/7
5,664,112 A * 9/1997 Sturgeon et al. 705/28
5,712,990 A * 1/1998 Henderson 705/28
5,726,884 A * 3/1998 Strugeon et al. 705/9
5,965,858 A * 10/1999 Suzuki et al. 235/375
6,067,549 A * 5/2000 Smalley et al. 707/104
6,097,995 A * 8/2000 Tipton et al. 700/266
6,122,622 A * 9/2000 Wiitala et al. 705/28
6,163,732 A * 12/2000 Petke et al. 700/106

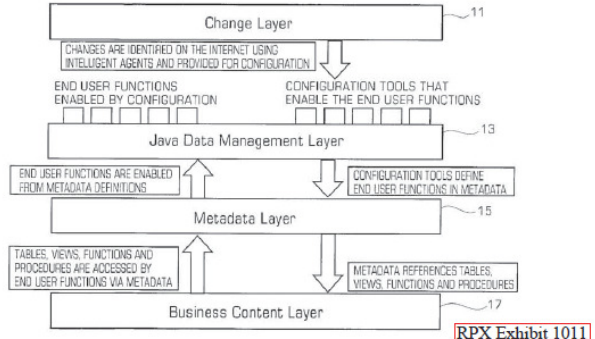
* cited by examiner

Primary Examiner—Thomas Black
Assistant Examiner—Mary Wang
(74) *Attorney, Agent, or Firm*—Gray Cary Ware & Freidenrich, LLP

(57) **ABSTRACT**

An integrated system for managing changes in regulatory and non-regulatory requirements for business activities at an industrial or commercial facility. Application of this system to environmental, health and safety activities, and to food, drug, cosmetic, and medical treatment and device activities, are discussed as examples. The system: provides one or more databases that contain information on operations and requirements concerning an activity or area of business; receives information on regulatory and non-regulatory changes that affect operations of the business; converts these changes into changes in data entry forms, data processing and analysis procedures, and presentation (by printing, electronic display and/or distribution) of data processing and analysis results to selected recipients, without requiring the services of one or more programmers to re-key and/or reformat the items affected by the change; and implements receipt of change information and dissemination of data processing and analysis results using the facilities of the Internet.

1 Claim, 13 Drawing Sheets



RPX Exhibit 1011
RPX v. AIT

What is claimed is:

1. A system for monitoring processing and disposition of at least one material used in a business at a facility, the system comprising:

a first database that provides product stewardship for at least one selected material that is received, created, consumed or produced as a waste product at the facility, the first data base including information on at least one product produced at the facility, information on ecological and toxicological studies performed at the facility, information for production of a Materials Safety Data Sheet (MSDS) for at least one material used at the facility, and tracking of allegations and inquiries concerning operations at the facility; and

at least one additional database, drawn from the following group of databases:

a second database that allows tracking and prevention of selected incidents involving unintended discharge of a material at the facility, the second database including information on response to at least one type of emergency at the facility, information on tracking of at least one incident at the facility, and safety information on at least one process used at the facility;

a third database that provides information on and monitoring of personnel health and safety at the facility, the third database including information on demographics of personnel working at the facility, information on safety measures implemented at the facility, injuries and illnesses experienced by at least one worker at the facility, and information on industrial hygiene and occupational medicine studies carried out at the facility;

a fourth database that provides information on and monitoring of hazardous materials and hazardous waste, the fourth database including information on at least one hazardous material used at the facility, tracking of at least one waste material produced at the facility, information on at pollution prevention measures and on site remediation measures implemented at the facility;

a fifth database that tracks a controlled release or discharge of a material to the environment, the fifth database including information on discharge of at least one hazardous substance into at least one of the air, the water, the groundwater and the soil at the facility, and information on at least one toxic chemical release at the facility;

a sixth database that provides selected information on regulatory requirements for receiving, handling, processing or producing hazardous materials, the sixth database including information on at least one environmental audit conducted at the facility, information on regulatory lists used at, and on regulatory issues concerning, the facility, and information on at least litigation issue concerning the facility; and

a seventh database that provides selected information on management of the facility, the seventh database including information on at least one of the physical structure and the organizational structure at the facility, information on tracking of at least one equipment item at the facility, and information on at least one process used at the facility;

a tools module that provides software for at least one of creation of a report on operations at the facility, creation of formulas and expressions for a report on operations, creation of at least one image for a report on operations, archiving of at least one record on operations, and security measures implemented at the facility, and that implements entry of one or more changes in regulatory and non-regulatory requirements for the business without requiring manual reprogramming of the tools module software; and

a relational database management module that links each database to each other database and to the tools module so that an information item, once entered, becomes available to each database and to the tools module.

* * * * *

Ex. 1011 in IPR2015-01750; Ex. 1013 in IPR2015-01751; Ex. 1113 in IPR2015-01752

14. (New) A system, comprising:

a server accessible by a browser executed on a client device, the server including a first portion, a second portion, a third portion, and a fourth portion,

the first portion of the server having information about unique aspects of a particular application,

the second portion of the server having information about user interface elements and one or more functions common to various applications, the various applications including the particular application,

the third portion of the server being configured to dynamically generate a functionality and a user interface for the particular application, the functionality and the user interface of the particular application being based on the information in the first portion of the server and the information in the second portion of the server, the third portion of the server being configured to send the functionality and the user interface for the particular application to the browser upon establishment of the connection between the server and the client,

the fourth portion of the server being configured to automatically detect changes that affect the information in the first portion of the server and the information in the second portion of the server.

Ex. 1012 at 5 in IPR2015-01750 – from 12/098,154

14. (New) A system, comprising:

a server accessible by a browser executed on a client device, the server including a first portion, a second portion, a third portion, and a fourth portion,

the first portion of the server having information about unique aspects of a particular application,

the second portion of the server having information about user interface elements and one or more functions common to various applications, the various applications including the particular application,

the third portion of the server being configured to dynamically generate a functionality and a user interface for the particular application, the functionality and the user interface of the particular application being based on the information in the first portion of the server and the information in the second portion of the server, the third portion of the server being configured to send the functionality and the user interface for the particular application to the browser upon establishment of the connection between the server and the client,

the fourth portion of the server being configured to automatically detect changes that affect the information in the first portion of the server and the information in the second portion of the server.

Ex. 1012 at 5 in IPR2015-01750 – from 12/098,154

13. A system, comprising:

a server accessible by a browser executed on a client device, the server including a first portion, a second portion, a third portion, and a fourth portion,

the first portion of the server having information about unique aspects of a particular application,

the second portion of the server having information about user interface elements and one or more functions common to various applications, the various applications including the particular application,

the third portion of the server being configured to dynamically generate a functionality and a user interface for the particular application, the functionality and the user interface of the particular application being based on the information in the first portion of the server and the information in the second portion of the server, the third portion of the server being configured to send the functionality and the user interface for the particular application to the browser upon establishment of a connection between the server and the client device,

the fourth portion of the server being configured to automatically detect changes that affect the information in the first portion of the server or the information in the second portion of the server.

Ex. 1001 at 33:19–34:8 in IPR2015-01750

14. (New) A system, comprising:

a server accessible by a browser executed on a client device, the server including a first portion, a second portion, a third portion, and a fourth portion,

the first portion of the server having information about unique aspects of a particular application,

the second p

or more functions

particular applicatio

the third po

and a user interface

particular applicatio

information in the s

send the functionali

establishment of the

the fourth p

affect the informati

of the server.

Ex. 1012 at 5 in IPR2015-01750 – from 12/098,154

Examiner Note

If the examiner had not read the application number she never would have realized that these claims went with this specification. The claims are extremely generic and broad there is no mention about regulatory changes or anything that the invention talks about in the first 13 pages of the specification that the invention is trying to solve/ directed at.

Currently the independent claim has 1) "unique aspect" and 2) "user interface element" for a particular application. If you talked to any software developer every project they worked on has at least these two elements and probably 100% of their software projects.

Ex. 1013 at 7-8 in IPR2015-01750

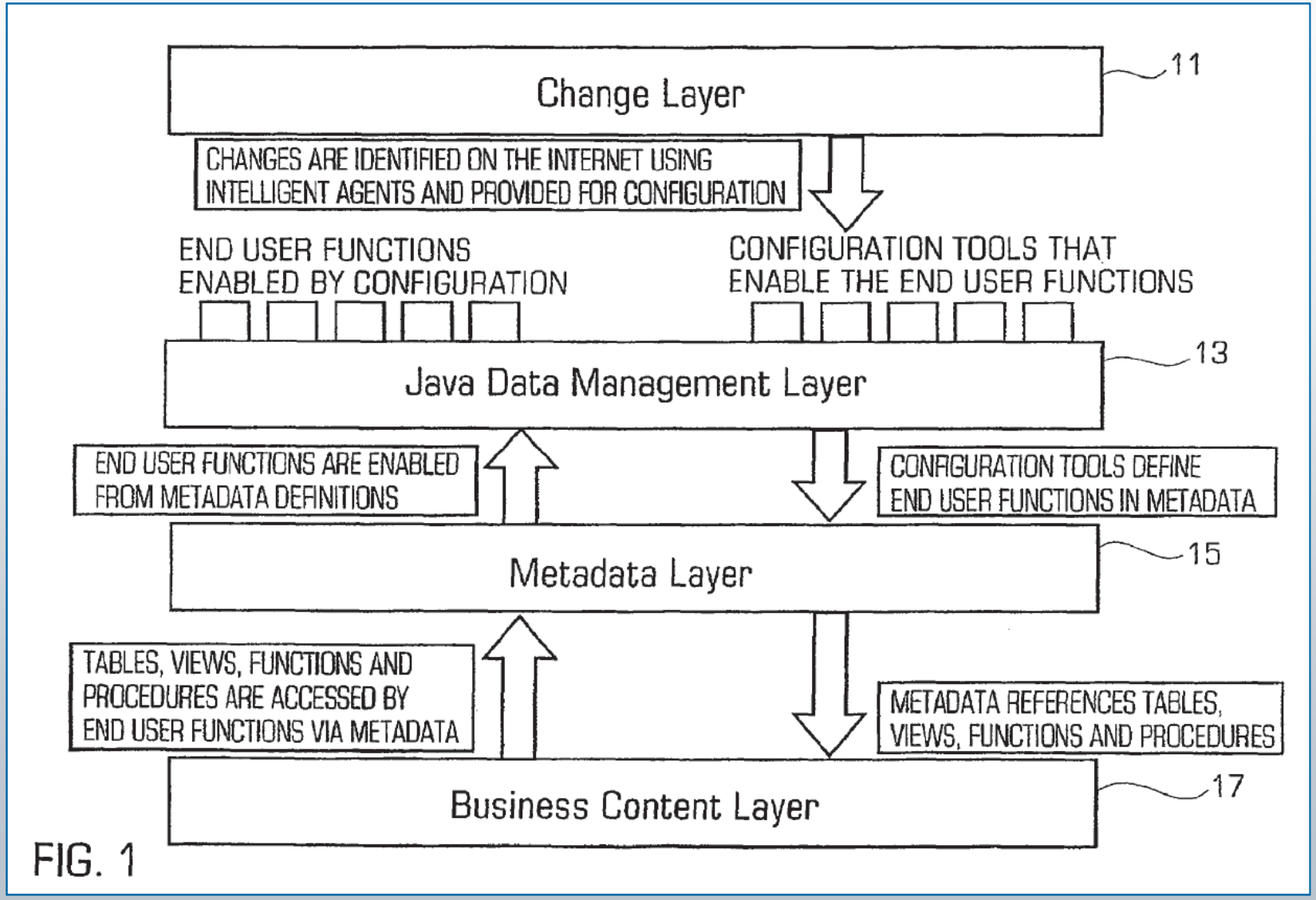
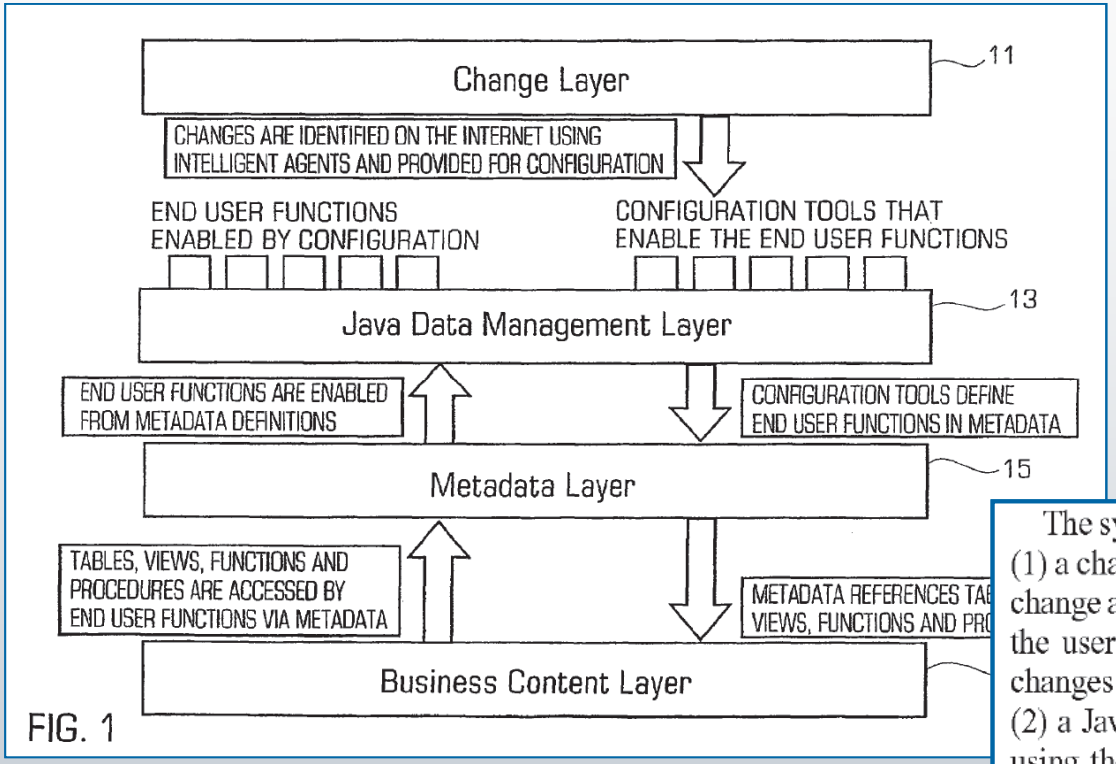


FIG. 1

Ex. 1001/1101 FIG. 1



Ex. 1001/1101 FIG. 1

The system operates at four layers, as illustrated in FIG. **1**: (1) a change management layer **11** that includes one or more change agents that "cruise the Web" and identify and bring to the user's attention relevant regulatory and non-regulatory changes found on the Web that may affect a user's business; (2) a Java data management layer **13**, a user interface, built using the Java language, that applies metadata attributes to business and business-change related data (regulation-based or non-regulation-based); (3) a metadata layer **15** that provides and/or defines data about every feature of the user interface including, without limitation, tools, worklists, data entry forms, reports, documents, processes, formulas, images, tables, views, columns, and other structures and functions; and (4) a business content layer **17** that is specific to the particular business operations of interest to the user.

Ex. 1001 at 9:38-52 in IPR2015-01750

Ex. 1001/1101 at 9:33-48 in IPR2015-01751 and -01752

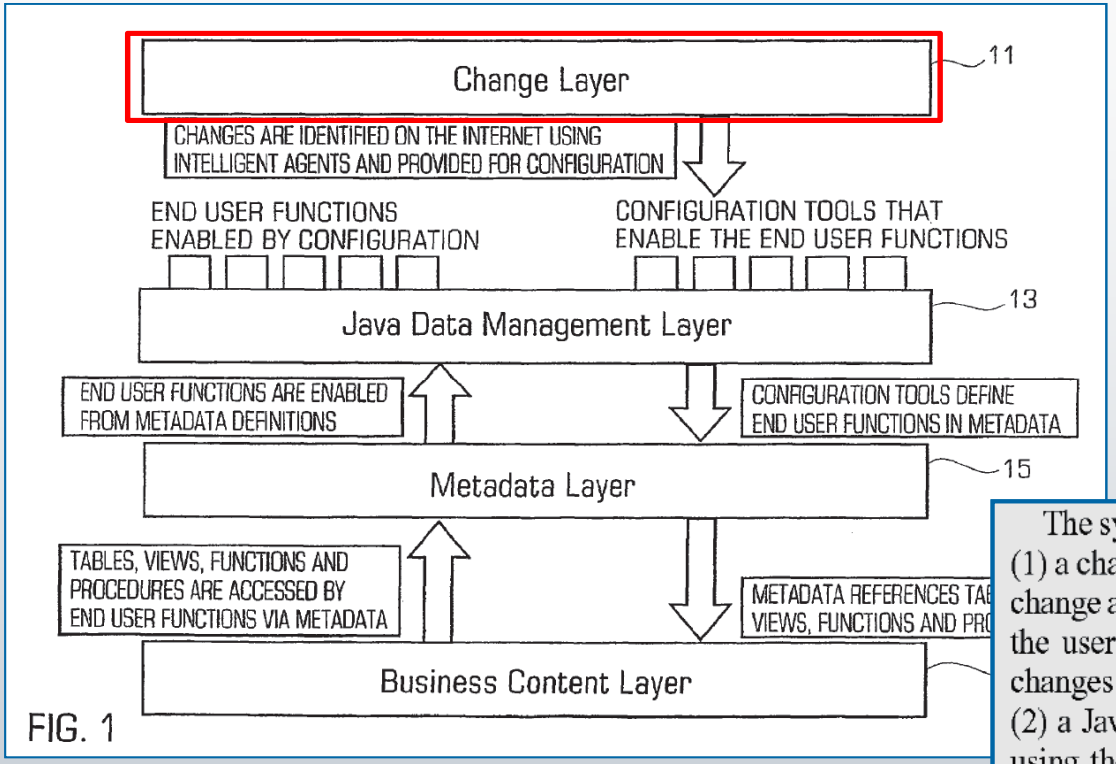


FIG. 1

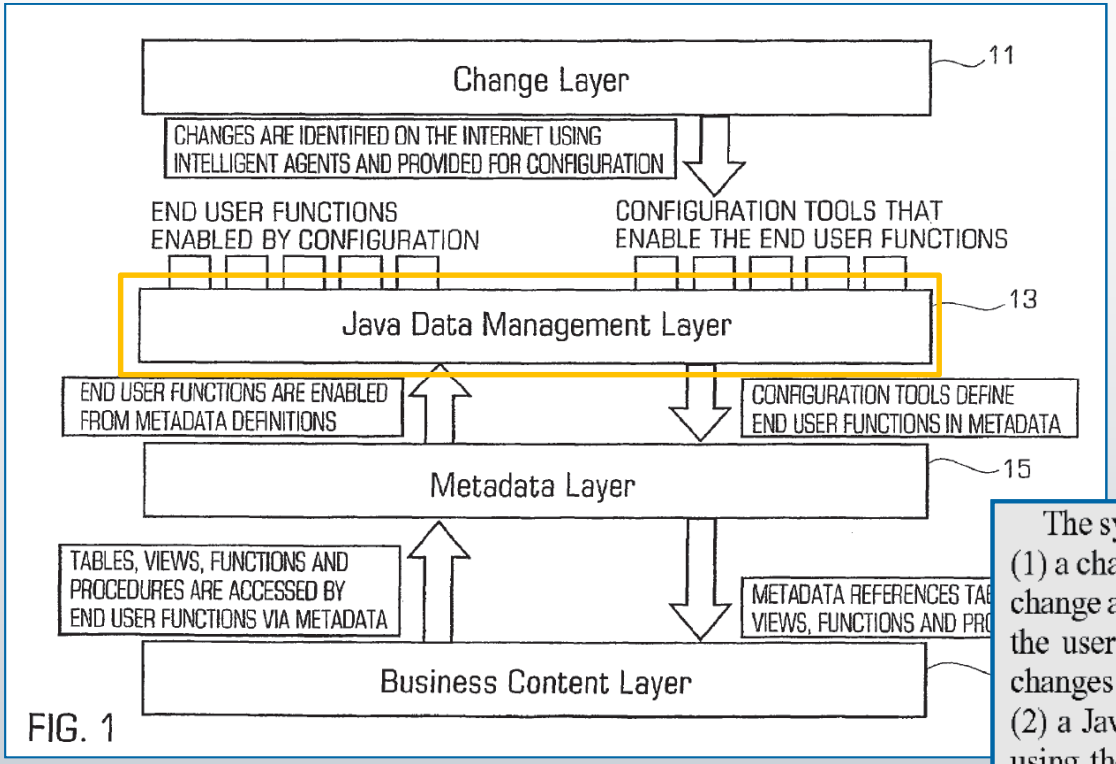
Ex. 1001/1101 FIG. 1

The system operates at four layers, as illustrated in FIG. **1**:
(1) a change management layer **11** that includes one or more change agents that "cruise the Web" and identify and bring to the user's attention relevant regulatory and non-regulatory changes found on the Web that may affect a user's business;
(2) a Java data management layer **13**, a user interface, built using the Java language, that applies metadata attributes to

(1) a change management layer **11** that includes one or more change agents that "cruise the Web" and identify and bring to the user's attention relevant regulatory and non-regulatory changes found on the Web that may affect a user's business;

particular business operations of interest to the user.

Ex. 1001 at 9:38-52 in IPR2015-01750
Ex. 1001/1101 at 9:33-48 in IPR2015-01751 and -01752



Ex. 1001/1101 FIG. 1

The system operates at four layers, as illustrated in FIG. **1**: (1) a change management layer **11** that includes one or more change agents that "cruise the Web" and identify and bring to the user's attention relevant regulatory and non-regulatory changes found on the Web that may affect a user's business; (2) a Java data management layer **13**, a user interface, built using the Java language, that applies metadata attributes to business and business-change related data (regulation-based

(2) a Java data management layer **13**, a user interface, built using the Java language, that applies metadata attributes to business and business-change related data (regulation-based or non-regulation-based);

Ex. 1001 at 9:38-52 in IPR2015-01750

Ex. 1001/1101 at 9:33-48 in IPR2015-01751 and -01752

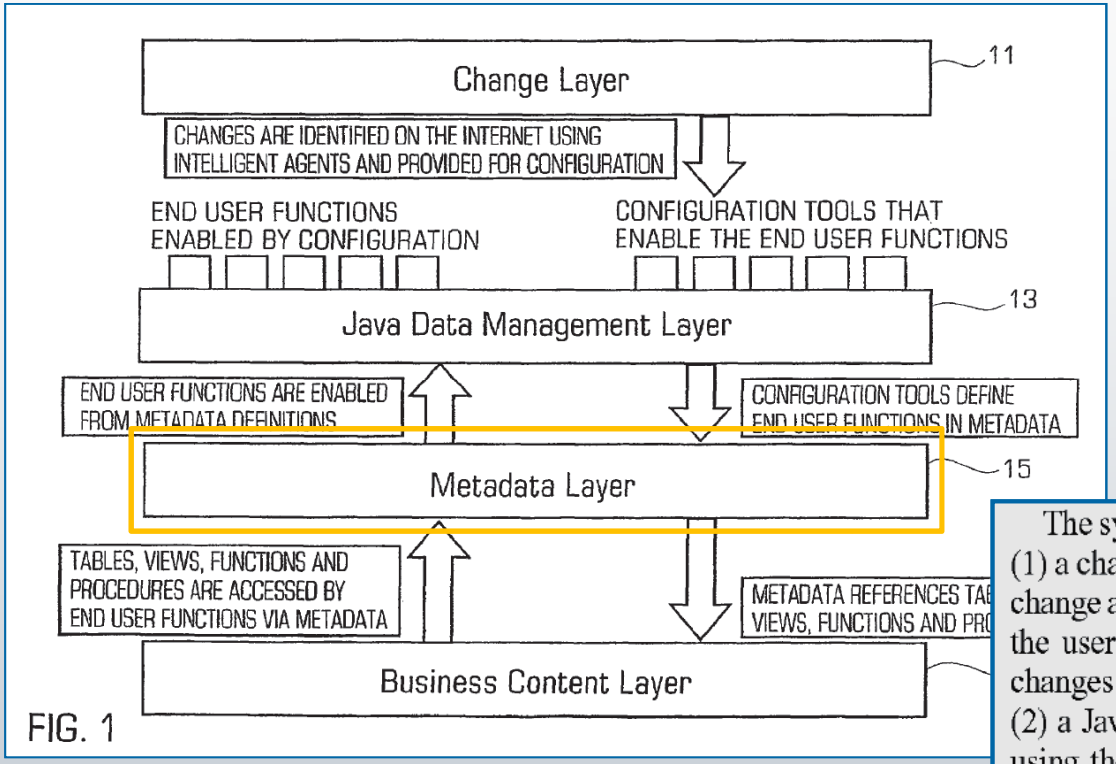


FIG. 1

Ex. 1001/1101 FIG. 1

(3) a metadata layer **15** that provides and/or defines data about every feature of the user interface including, without limitation, tools, worklists, data entry forms, reports, documents, processes, formulas, images, tables, views, columns, and other structures and functions;

The system operates at four layers, as illustrated in FIG. **1**: (1) a change management layer **11** that includes one or more change agents that "cruise the Web" and identify and bring to the user's attention relevant regulatory and non-regulatory changes found on the Web that may affect a user's business; (2) a Java data management layer **13**, a user interface, built using the Java language, that applies metadata attributes to data (regulation-based ata layer **15** that pro- y feature of the user , tools, worklists, data processes, formulas, er structures and func- **7** that is specific to the particular business operations of interest to the user.

Ex. 1001 at 9:38-52 in IPR2015-01750

Ex. 1001/1101 at 9:33-48 in IPR2015-01751 and -01752

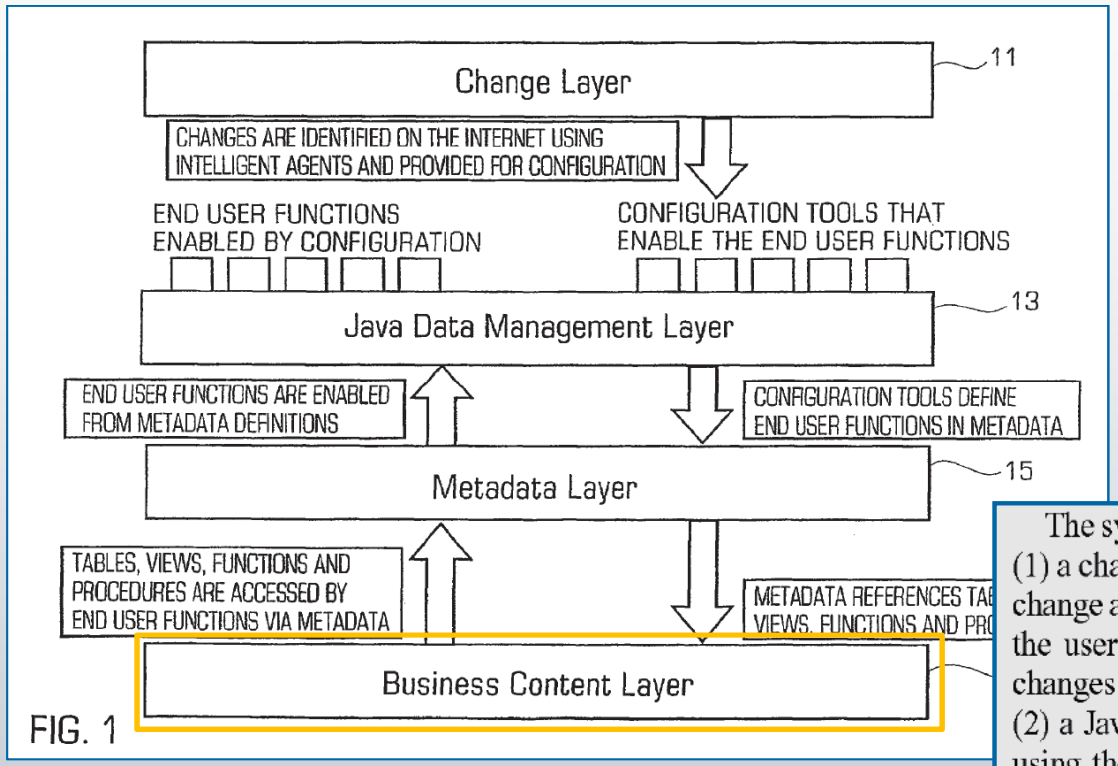


FIG. 1

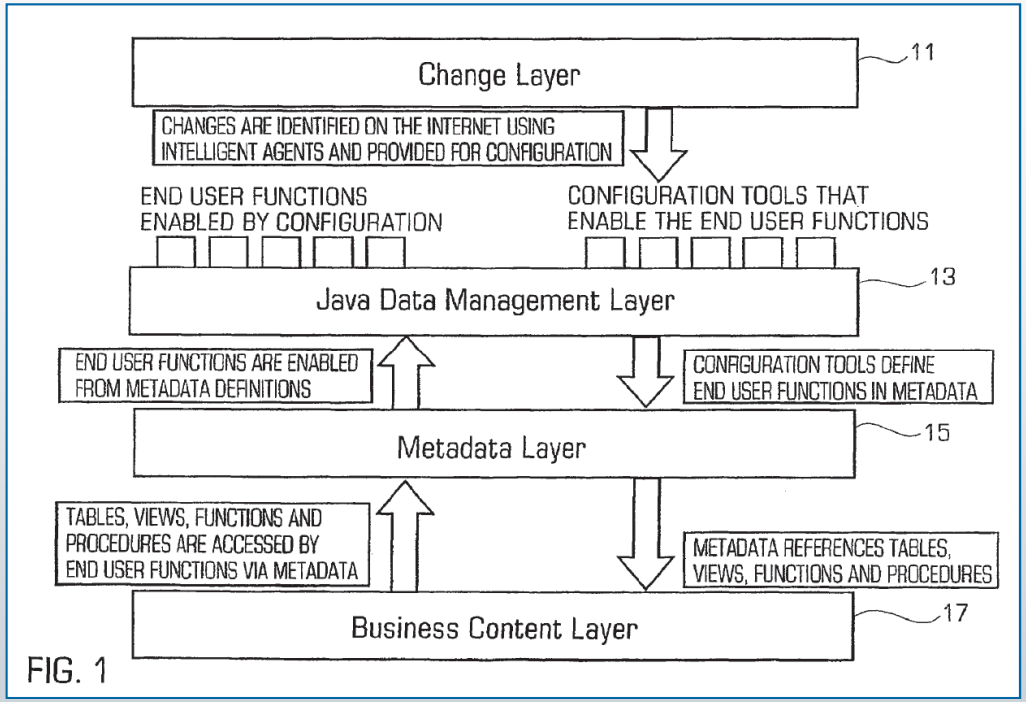
Ex. 1001/1101 FIG. 1

The system operates at four layers, as illustrated in FIG. **1**: (1) a change management layer **11** that includes one or more change agents that "cruise the Web" and identify and bring to the user's attention relevant regulatory and non-regulatory changes found on the Web that may affect a user's business; (2) a Java data management layer **13**, a user interface, built using the Java language, that applies metadata attributes to business and business-change related data (regulation-based ro- ser ata entry forms, reports, documents, processes, formulas, images, tables, views, columns, and other structures and functions; and (4) a business content layer **17** that is specific to the particular business operations of interest to the user.

(4) a business content layer **17** that is specific to the particular business operations of interest to the user.

Ex. 1001 at 9:38-52 in IPR2015-01750

Ex. 1001/1101 at 9:33-48 in IPR2015-01751 and -01752



Ex. 1001/1101 FIG. 1

1. A system for providing a dynamically generated application having one or more functions and one or more user interface elements; comprising:

a server computer;

one or more client computers connected to the server computer over a computer network;

a first layer associated with the server computer containing information about the unique aspects of a particular application;

a second layer associated with the server computer containing information about the user interface and functions common to a variety of applications, a particular application being generated based on the data in both the first and second layers;

a third layer associated with the server computer that retrieves the data in the first and second layers in order to generate the functionality and user interface elements of the application; and

a change management layer for automatically detecting changes that affect an application,

each client computer further comprising a browser application being executed by each client computer, wherein a user interface and functionality for the particular application is distributed to the browser application and dynamically generated when the client computer connects to the server computer.

Ex. 1001/1101 at 32:9-34 in IPR2015-01751 and -01752

II. AIT'S SOLE BASIS FOR CHALLENGING THE GROUNDS IS THAT THE PRIOR ART ALLEGEDLY DOES NOT TEACH "AUTOMATICALLY DETECTING CHANGES THAT AFFECT AN APPLICATION"

For each of Grounds 1-4, the **only** limitation that the POR and AIT's supporting expert declarations alleged was not met was "automatically detecting changes that affect a particular application" in claim 21 and the change management layer for performing that function in claim 1[3]. POR at 22-32; Ex. 2032 ¶¶51-84; Ex. 2033 ¶¶41-64; Ex. 1058 at 118:3-119:6, 127:5-128:8, 143:2-144:5 (Dr. Jagadish conceded that he offered no opinion that the prior art fails to meet any other claim limitation)[4].

Petitioner's Reply to POR, Paper 72 at 4 in IPR2015-01751

1. A system for providing a dynamically generated application having one or more functions and one or more user interface elements; comprising:

a server computer;

one or more client computers connected to the server computer over a computer network;

a first layer associated with the server computer containing information about the unique aspects of a particular application;

a second layer associated with the server computer containing information about the user interface and functions common to a variety of applications, a particular application being generated based on the data in both the first and second layers;

a third layer associated with the server computer that retrieves the data in the first and second layers in order to generate the functionality and user interface elements of the application; and

a change management layer for automatically detecting changes that affect an application,

each client computer further comprising a browser application being executed by each client computer, wherein a user interface and functionality for the particular application is distributed to the browser application and dynamically generated when the client computer connects to the server computer.

'482 Patent (Ex. 1001), claim 1.

a change management layer for automatically detecting changes that affect an application,

27. In my opinion, the broadest reasonable interpretation a POSITA would apply to a "change management layer" is automatically detecting changes which impact how the application program should operate. In the context of the '482 patent, these "changes" detected by the change management layer arise from changes external to the application program.

RPX Exhibit 1057
RPX v. AIT
IPR2015-01751

RPX Exhibit 1057
RPX v. AIT

Jagadish Decl. (Ex. 2032) at ¶ 27.

> **a change management layer for automatically detecting changes that affect an application**,
>
> each client computer further comprising a browser application being executed by each client computer, wherein a user interface and functionality for the particular application is distributed to the browser application and dynamically generated when the client computer connects to the server computer.

Both the Board implicitly and Petitioner explicitly define "change management layer" to read on detecting changes *internal* to an application program. Yet, detecting changes *internal* to an application program is precisely what the claimed "third layer" does. A POSITA clearly would recognize this. (*see, e.g.* Ex. 2032, ¶¶ 34-36)

The term "change management layer" is not a term of art. However, the term "change management layer" when interpreted in view of the specification would readily be understood to a person of ordinary skill in the art to mean "a layer that automatically detects changes external to the application program which impact how the application program should operate." (Ex. 2032, ¶ 27; Ex. 2033, ¶ 27) The '482 patent includes an extensive description of the change management layer, sometimes referred to as the "change layer:" In particular, the '482 patent states (col. 9, lines 33-38):

> The system operates at four layers, as illustrated in FIG. 1: (1) **a change management layer 11 that includes one or**

POR, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751

Q. Right. What I'm trying to get at with my question is that what your construction does is construe the functional words that follow a change management layer for; right? Those words are automatically detecting changes that affect an application. So you're offering a construction of what that function means; right?

A. Yes. And I think what I was trying to say was I'm not trying to limit myself to some kind of functional construction. I'm trying to construe change management layer and yes, it is true that my understanding of change management layer is to a large extent determined by the functions that such a layer should perform.

Q. Well, when you say such a layer should perform, the claim explicitly says what function it performs; right?

A. Well, that's the best intrinsic evidence in support of my understanding.

Q. And you have construed the function in the claim as meeting your construction? Sorry. That's a terrible question.

You have construed the words "for automatically detecting changes that affect an application"? That is what you offered a construction

of, not the words "change management layer"; right?

A. Yeah. And I think that that's what I'm trying to explain. I think that to the extent change management layer is not a term of art, when one attempts to understand what that could possibly mean in the context of these claims, one has to look at the claims and the spec and whatever else one knows that's relevant -- or one at that time would have known that's relevant, and I am agreeing with you to the extent you're saying that the definition of change management layer is to a large extent already there in the claim language itself in terms of the function that it performs as stated in the claim itself. That is consistent with everything else that we know about change management layer with respect to what is stated about it in the spec and elsewhere.

Q. Okay. So can I point you to Paragraphs 42 and 43 of your declaration?

A. Yes.

Q. So in these paragraphs -- what Dr. Crovella said was I believe exactly what you just said, which is that when you construe that clause "change management layer" performing the function, it should be construed to be a layer that performs the function explicitly recited in the claim, and Paragraphs 42 and 43 of your

AIT's Expert, Ex. 1058/1158 at 102:9-13

1. A system for providing a dynamically generated application having one or more functions and one or more user interface elements; comprising:

a server computer;

one or more client computers connected to the server computer over a computer network;

a first layer associated with the server computer containing information about the unique aspects of a particular application;

a second layer associated with the server computer containing information about the user interface and functions common to a variety of applications, a particular application being generated based on the data in both the first and second layers;

a third layer associated with the server computer that retrieves the data in the first and second layers in order to generate the functionality and user interface elements of the application; and

a change management layer for automatically detecting changes that affect an application,

each client computer further comprising a browser application being executed by each client computer, wherein a user interface and functionality for the particular application is distributed to the browser application and dynamically generated when the client computer connects to the server computer.

'482 Patent (Ex. 1001), claim 1.

a change management layer for automatically detecting changes that affect an application,

27. In my opinion, the broadest reasonable interpretation a POSITA would apply to a "change management layer" is automatically detecting changes which impact how the application program should operate. In the context of the '482 patent, these "changes" detected by the change management layer arise from changes external to the application program.

RPX Exhibit 1057
RPX v. AIT
IPR2015-01751

RPX Exhibit 1057
RPX v. AIT

Jagadish Decl. (Ex. 2032) at ¶ 27.

a change management layer for automatically detecting changes that affect an application,

27. In my opinion, the broadest reasonable interpretation a POSITA would apply to a "change management layer" is automatically detecting changes which impact how the application program should operate. In the context of the '482 patent, these "changes" detected by the change management layer arise from changes external to the application program.

RPX Exhibit 1057
RPX v. AIT
IPR2015-01750

RPX Exhibit 1057
RPX v. AIT

Jagadish Decl. (Ex. 2032) at ¶ 27.

Ex. 1057/1157

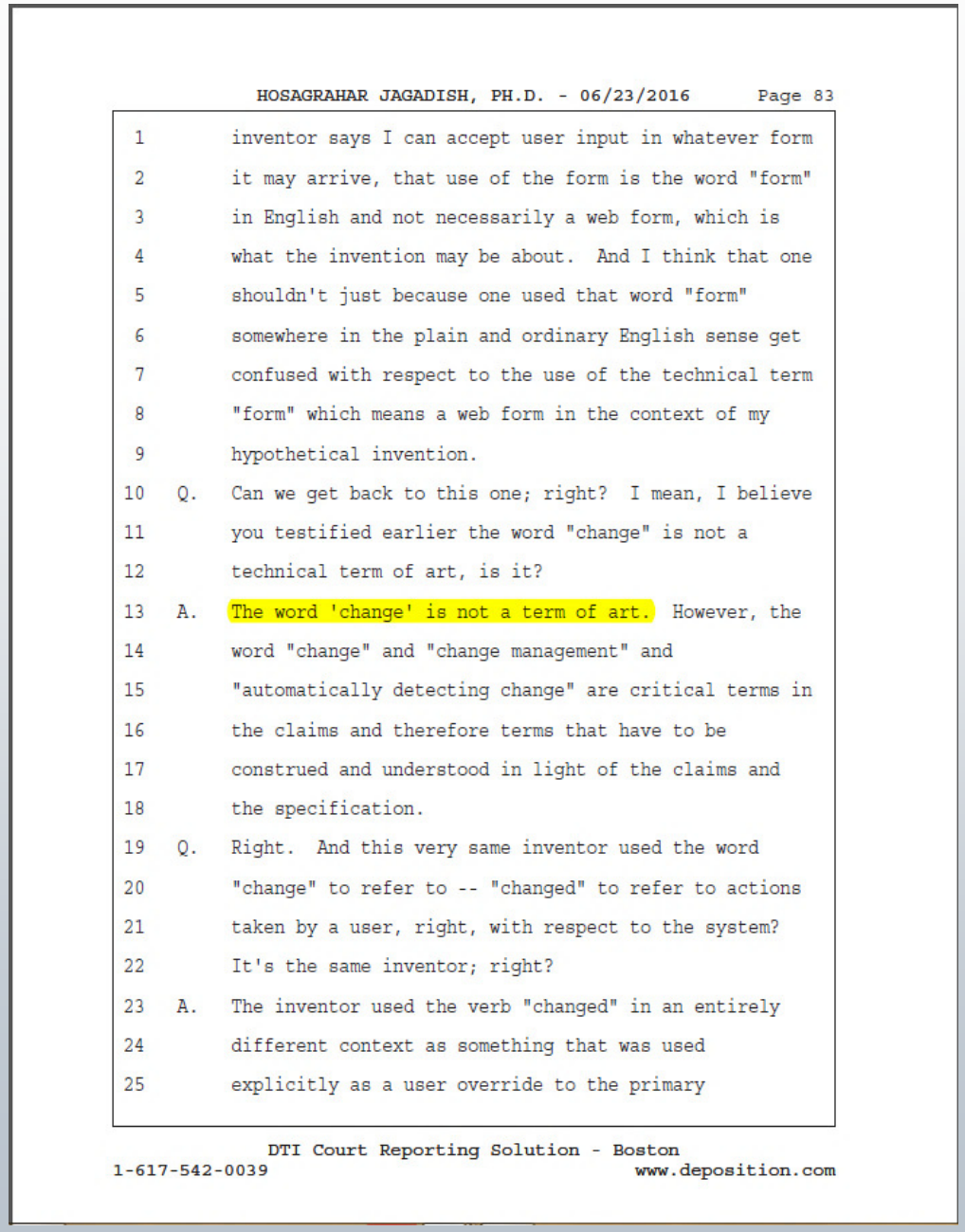
Q. Okay. So I want to show you another exhibit that we've marked as 1057, and it is the same exhibit but we've color coded the words I think we've just walked through and agreed how they correspond to each other. So I'd like to just quickly walk through this. So the green boxes illustrate that the words "automatically detecting" in the claim appear in the construction verbatim; right?

A. Yes.

Q. And the yellow boxes illustrate that the word "changes" in the claim is replaced in the construction by "'changes' detected by the change management layer arise from changes external to the application program"; right?

A. Yes.

AIT's Expert, Ex. 1058/1158 at 94:1-15



HOSAGRAHAR JAGADISH, PH.D. - 06/23/2016 Page 83

inventor says I can accept user input in whatever form it may arrive, that use of the form is the word "form" in English and not necessarily a web form, which is what the invention may be about. And I think that one shouldn't just because one used that word "form" somewhere in the plain and ordinary English sense get confused with respect to the use of the technical term "form" which means a web form in the context of my hypothetical invention.

Q. Can we get back to this one; right? I mean, I believe you testified earlier the word "change" is not a technical term of art, is it?

A. The word 'change' is not a term of art. However, the word "change" and "change management" and "automatically detecting change" are critical terms in the claims and therefore terms that have to be construed and understood in light of the claims and the specification.

Q. Right. And this very same inventor used the word "change" to refer to -- "changed" to refer to actions taken by a user, right, with respect to the system? It's the same inventor; right?

A. The inventor used the verb "changed" in an entirely different context as something that was used explicitly as a user override to the primary

DTI Court Reporting Solution - Boston
1-617-542-0039 www.deposition.com

AIT's Expert, Ex. 1058/1158 at 83:13

> **a change management layer for automatically detecting changes that affect an application,**
>
> each client computer further comprising a browser application being executed by each client computer, wherein a user interface and functionality for the particular application is distributed to the browser application and dynamically generated when the client computer connects to the server computer.

Both the Board implicitly and Petitioner explicitly define "change management layer" to read on detecting changes *internal* to an application program. Yet, detecting changes *internal* to an application program is precisely what the claimed "third layer" does. A POSITA clearly would recognize this. (*see, e.g.* Ex. 2032, ¶¶ 34-36)

The term "change management layer" is not a term of art. However, the term "change management layer" when interpreted in view of the specification would readily be understood to a person of ordinary skill in the art to mean "a layer that automatically detects changes external to the application program which impact how the application program should operate." (Ex. 2032, ¶ 27; Ex. 2033, ¶ 27) The '482 patent includes an extensive description of the change management layer, sometimes referred to as the "change layer:" In particular, the '482 patent states (col. 9, lines 33-38):

> The system operates at four layers, as illustrated in FIG. 1: (1) **a change management layer 11 that includes one or**

> **more change agents that "cruise the Web" and identify and bring to the user's attention relevant regulatory and non-regulatory changes found on the Web that may affect a user's business**; (2) a Java data management layer 13, a user interface, built using the Java language, that applies metadata attributes to business and business-change related data (regulation-based or non-regulation-based); (3) a metadata layer 15 that provides and/or defines data about every feature of the user interface including, without limitation, tools, worklists, data entry forms, reports, documents, processes, formulas, images, tables, views, columns, and other structures and functions; and (4) a business content layer 17 that is specific to the particular business operations of interest to the user.

Figure 1 of the '482 patent shows a change layer that is separate from the three other layers of the system:

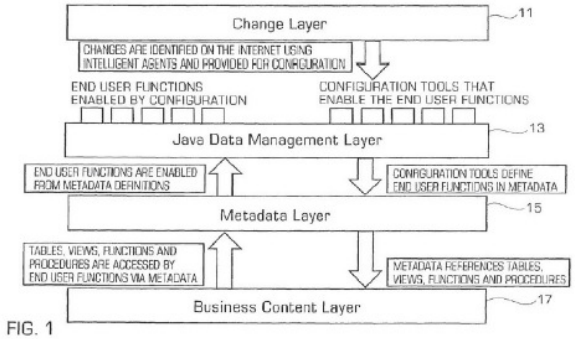


FIG. 1

POR at 14, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751

not changes that require automatic detection, and so if you read changes to include those things, then you end up with an inconsistent reading of the phrase "automatically detect changes."

Q. Well, let's talk about that. Because I think there's a difference between saying you end up with an interpretation that is not limited to the embodiment and an interpretation that is inconsistent; right?

A. Yes. I'm talking about inconsistency. I'm not -- I have never tried to limit the claims to the environment.

Q. So let's talk about that. Do you believe that Dr. Crovella's construction covers the embodiments in the spec?

A. Dr. Crovella's construction, being overly broad, would cover the embodiments in the spec because -- here's sort of the logic. Change can mean A or B. If change means A, let's say that it covers the embodiments of the spec. If change means B, it leads to an inconsistent interpretation of the spec -- of the -- of the claims. Sorry. The claims -- the claim language is inconsistent. I'm interpreting change to mean, A, he wants to interpret it to mean A or B. If he interprets it as A -- if I take his A or B and deconstruct it, if we interpret it as A, we're in

agreement, if we interpret it as B, then he's inconsistent, and I therefore declare his interpretation as A or B as overly broad and believe that the correct interpretation should be A and not A or B.

Q. Well, let me give you a hypothetical that maybe is simple. The specification describes an orange and the patent drafter calls it a fruit. Okay? Somebody in the petitioner's position challenge it and shows a reference that shows an apple. Is an apple a fruit?

A. An apple is a fruit.

Q. It's not the fruit in the specification?

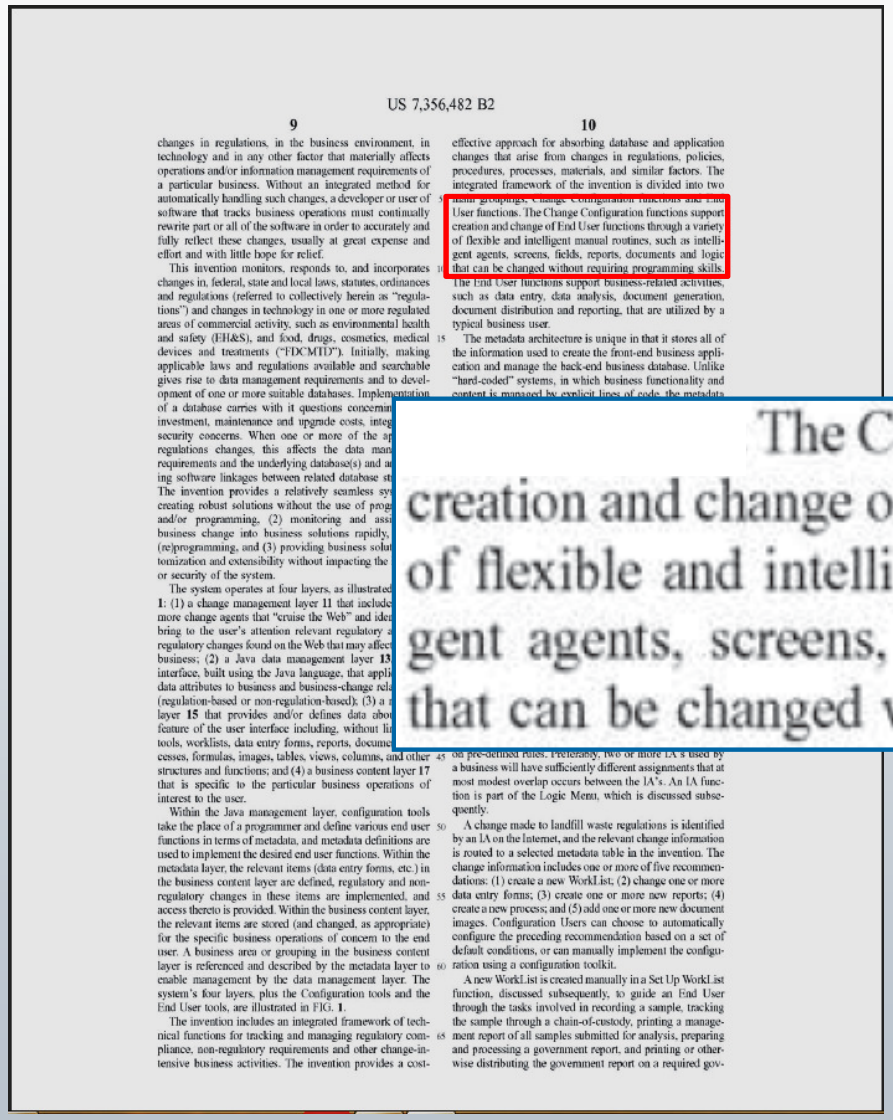
A. That is correct.

Q. Calling the -- interpreting the fruit broadly enough to cover an apple and an orange is not inconsistent with the specification; right?

A. No, it is not. In fact, that is -- that's exactly how one should be reading claims.

Q. Okay. So in our example I think we agreed that change that happens in some government database is the same type of change that happens to the internal database. What I understand you to be saying is, yeah, but the type of change that happens to the internal database is not the type of change that is detected by the embodiments in the specification and therefore I'm

AIT's Expert, Ex. 1058/1158 at 115:12-17



US 7,356,482 B2

9

changes in regulations, in the business environment, in technology and in any other factor that materially affects operations and/or information management requirements of a particular business. Without an integrated method for automatically handling such changes, a developer or user of software that tracks business operations must continually rewrite part or all of the software in order to accurately and fully reflect these changes, usually at great expense and effort and with little hope for relief.

This invention monitors, responds to, and incorporates changes in, federal, state and local laws, statutes, ordinances and regulations (referred to collectively herein as "regulations") and changes in technology in one or more regulated areas of commercial activity, such as environmental health and safety (EH&S), and food, drugs, cosmetics, medical devices and treatments ("FDCMTD"). Initially, making applicable laws and regulations available and searchable gives rise to data management requirements and to development of one or more suitable databases. Implementation of a database carries with it questions concernin investment, maintenance and upgrade costs, integ security concerns. When one or more of the ap regulations changes, this affects the data man requirements and the underlying database(s) and a ing software linkages between related database st The invention provides a relatively seamless sy creating robust solutions without the use of prog and/or programming, (2) monitoring and assi business change into business solutions rapidly, (re)programming, and (3) providing business solut tomization and extensibility without impacting the or security of the system.

The system operates at four layers, as illustrated 1: (1) a change management layer 11 that include more change agents that "cruise the Web" and iden bring to the user's attention relevant regulatory regulatory changes found on the Web that may affect business; (2) a Java data management layer 13 interface, built using the Java language, that appli data attributes to business and business-change rela (regulation-based or non-regulation-based); (3) a layer 15 that provides and/or defines data abou feature of the user interface including, without li tools, worklists, data entry forms, reports, docume cesses, formulas, images, tables, views, columns, and other structures and functions; and (4) a business content layer 17 that is specific to the particular business operations of interest to the user.

Within the Java management layer, configuration tools take the place of a programmer and define various end user functions in terms of metadata, and metadata definitions are used to implement the desired end user functions. Within the metadata layer, the relevant items (data entry forms, etc.) in the business content layer are defined, regulatory and non-regulatory changes in these items are implemented, and access thereto is provided. Within the business content layer, the relevant items are stored (and changed, as appropriate) for the specific business operations of concern to the end user. A business area or grouping in the business content layer is referenced and described by the metadata layer to enable management by the data management layer. The system's four layers, plus the Configuration tools and the End User tools, are illustrated in FIG. 1.

The invention includes an integrated framework of technical functions for tracking and managing regulatory compliance, non-regulatory requirements and other change-intensive business activities. The invention provides a cost-

10

effective approach for absorbing database and application changes that arise from changes in regulations, policies, procedures, processes, materials, and similar factors. The integrated framework of the invention is divided into two [illegible] groupings, Change Configuration functions and End User functions. The Change Configuration functions support creation and change of End User functions through a variety of flexible and intelligent manual routines, such as intelligent agents, screens, fields, reports, documents and logic that can be changed without requiring programming skills. The End User functions support business-related activities, such as data entry, data analysis, document generation, document distribution and reporting, that are utilized by a typical business user.

The metadata architecture is unique in that it stores all of the information used to create the front-end business application and manage the back-end business database. Unlike "hard-coded" systems, in which business functionality and content is managed by explicit lines of code, the metadata [illegible] on pre-defined rules. Preferably, two or more IA's used by a business will have sufficiently different assignments that at most modest overlap occurs between the IA's. An IA function is part of the Logic Menu, which is discussed subsequently.

A change made to landfill waste regulations is identified by an IA on the Internet, and the relevant change information is routed to a selected metadata table in the invention. The change information includes one or more of five recommendations: (1) create a new WorkList; (2) change one or more data entry forms; (3) create one or more new reports; (4) create a new process; and (5) add one or more new document images. Configuration Users can choose to automatically configure the preceding recommendation based on a set of default conditions, or can manually implement the configuration using a configuration toolkit.

A new WorkList is created manually in a Set Up WorkList function, discussed subsequently, to guide an End User through the tasks involved in recording a sample, tracking the sample through a chain-of-custody, printing a management report of all samples submitted for analysis, preparing and processing a government report, and printing or otherwise distributing the government report on a required gov-

The Change Configuration functions support creation and change of End User functions through a variety of flexible and intelligent manual routines, such as intelligent agents, screens, fields, reports, documents and logic that can be changed without requiring programming skills.

'482 patent at 10:7-10, Ex. 1001 in IPR2015-01751, Ex. 1101 in IPR2015-01752

That's what the invention's about.
Q. Well, with all due respect, you keep saying what the invention is about. I guess I'm asking the specification says you can change these things through a manual routine, which is user action; right?
A. The -- you can change end-user functions through manual actions and the -- and the specification explicitly allows for manual actions and explicitly allows for a manual override, as we were discussing previously.
Q. And the inventor uses the word "change" to describe something that it is directed by user interaction; right?
A. That is for the manual override, and I think the point is the word "change," again, because it's such a simple common word in the English language, becomes the changes that are -- the changes that are being detected, and the changes that are the changes of concern to the change management layer and the change configuration layer result in changes to the application software, changes to the metadata, and I think what we are looking at here is this latter category of change where there is a manual override, and yes, that is there and yes, there is a manual override.

AIT's Expert, Ex. 1058/1158 at 86:11-25

Claim Term / Phrase	AIT Proposed Construction	Salesforce Proposed Construction
"changes that affect a particular application"/ "changes that affect an application" ('482 claims 1, 21)	"changes to an application's metadata"	"modifications to regulatory, technological, or social requirements stored in a third party repository that affect an application"

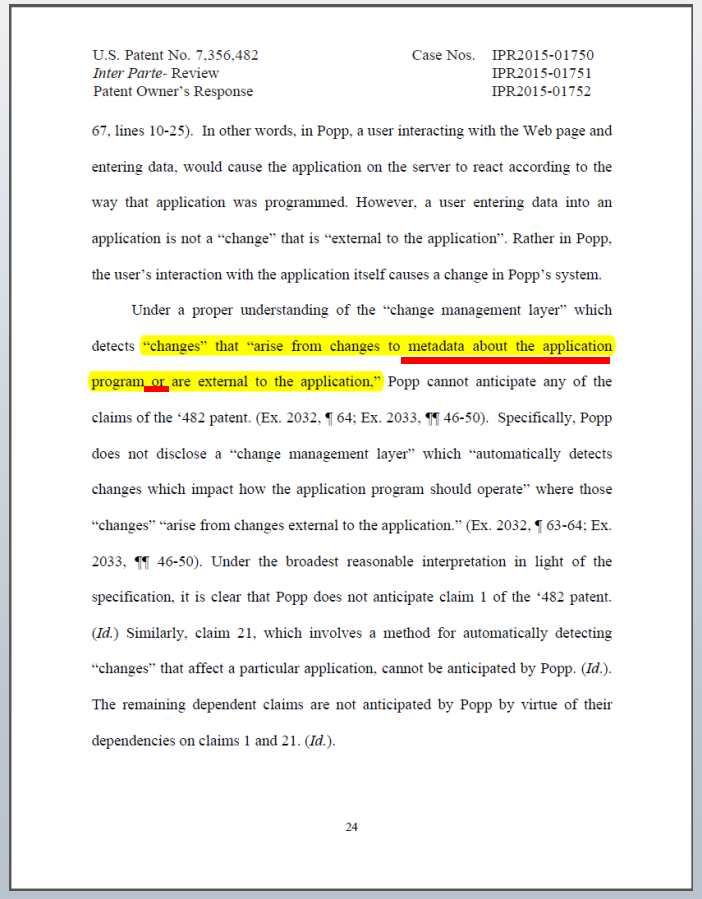
Ex. 1059/1159 at 11

> The term "change management layer" would be understood to one of ordinary skill in the art as "a layer that automatically detects changes which impact how the application program should operate." (Ex. 2032, ¶ 27; Ex. 2033, ¶ 26) The associated "changes" "arise from changes external to the application program." (Ex. 2032, ¶ 27; Ex. 2033, ¶¶ 27-28)

POR at 18, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751

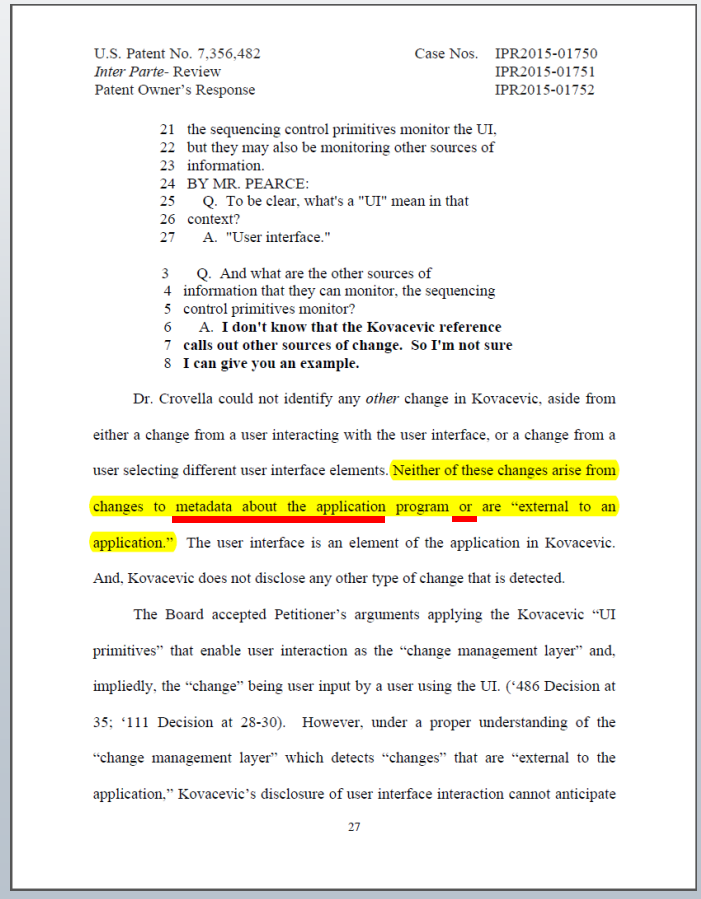
The term "change management layer" would be understood to one of ordinary skill in the art as "a layer that automatically detects changes which impact how the application program should operate." (Ex. 2032, ¶ 27; Ex. 2033, ¶ 26) The associated "changes" "arise from changes external to the application program." (Ex. 2032, ¶ 27; Ex. 2033, ¶¶ 27-28)

POR at 18, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751



67, lines 10-25). In other words, in Popp, a user interacting with the Web page and entering data, would cause the application on the server to react according to the way that application was programmed. However, a user entering data into an application is not a "change" that is "external to the application". Rather in Popp, the user's interaction with the application itself causes a change in Popp's system.

Under a proper understanding of the "change management layer" which detects "changes" that "arise from changes to metadata about the application program or are external to the application," Popp cannot anticipate any of the claims of the '482 patent. (Ex. 2032, ¶ 64; Ex. 2033, ¶¶ 46-50). Specifically, Popp does not disclose a "change management layer" which "automatically detects changes which impact how the application program should operate" where those "changes" "arise from changes external to the application." (Ex. 2032, ¶ 63-64; Ex. 2033, ¶¶ 46-50). Under the broadest reasonable interpretation in light of the specification, it is clear that Popp does not anticipate claim 1 of the '482 patent. (*Id.*) Similarly, claim 21, which involves a method for automatically detecting "changes" that affect a particular application, cannot be anticipated by Popp. (*Id.*). The remaining dependent claims are not anticipated by Popp by virtue of their dependencies on claims 1 and 21. (*Id.*).



the sequencing control primitives monitor the UI, but they may also be monitoring other sources of information.
BY MR. PEARCE:
Q. To be clear, what's a "UI" mean in that context?
A. "User interface."

Q. And what are the other sources of information that they can monitor, the sequencing control primitives monitor?
A. **I don't know that the Kovacevic reference calls out other sources of change. So I'm not sure I can give you an example.**

Dr. Crovella could not identify any *other* change in Kovacevic, aside from either a change from a user interacting with the user interface, or a change from a user selecting different user interface elements. Neither of these changes arise from changes to metadata about the application program or are "external to an application." The user interface is an element of the application in Kovacevic. And, Kovacevic does not disclose any other type of change that is detected.

The Board accepted Petitioner's arguments applying the Kovacevic "UI primitives" that enable user interaction as the "change management layer" and, impliedly, the "change" being user input by a user using the UI. ('486 Decision at 35; '111 Decision at 28-30). However, under a proper understanding of the "change management layer" which detects "changes" that are "external to the application," Kovacevic's disclosure of user interface interaction cannot anticipate

POR at 24 and 27, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751

patent lacks merit. The specification does not state that Figure 1 depicts the sum total of the patentee's invention. To the contrary, the specification simply states that Figure 1 "schematically *illustrates* the relationship of four layers that are the primary components of the invention." (Boebel Decl., Ex. 1 ('482 patent, at 8:50-51)). Thus, the specification makes clear that Figure 1 is merely a high-level depiction of the general relationship between the four layers of the disclosed invention and does not limit the scope of the asserted claims. *See Innova/Pure, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1117 (Fed. Cir. 2004) ("particular embodiments appearing in the written description will not be used to limit claim language that has broader effect[.] . . . unless the patentee has demonstrated a clear intention to limit the claim scope using words or expressions of manifest exclusion or restriction.") (internal quotations and citations omitted).

Salesforce also erroneously contends that the "changes that affect . . ." limitations should be limited to three specific categories of "modifications to regulatory, technological, or social requirements." Salesforce asserts that "the specification does not identify any other categories of material changes detected by the claimed change management layer," but this is incorrect. (Def. Br. at 20:8-13). The specification states that the change management layer "includes one or more change agents that . . . identify and bring to the user's attention relevant regulatory *and non-regulatory changes* found on the Web that may affect a user's business." (Boebel Decl., Ex. 1 ('482 patent, at 9:34-38)). In other words, the specification describes that the change management layer can detect any type of change that may have an impact on the user's business, not just changes within certain categories of subject matter.

C. "dynamically generate . . ."

Claim Term / Phrase	AIT Proposed Construction	Salesforce Proposed Construction
"dynamically generate a functionality and a user interface" ('111 claim 13)	"dynamically generate" means "generate or update when needed." No construction necessary for "a functionality and a user interface."	Indefinite, or in the alternative, requiring at least "generate [both a functionality and a user interface] immediately and concurrently without any modification of software by a user"

Ex. 1060/1160 at 7

II. AIT'S SOLE BASIS FOR CHALLENGING THE GROUNDS IS THAT THE PRIOR ART ALLEGEDLY DOES NOT TEACH "AUTOMATICALLY DETECTING CHANGES THAT AFFECT AN APPLICATION"

For each of Grounds 1-4, the **only** limitation that the POR and AIT's supporting expert declarations alleged was not met was "automatically detecting changes that affect a particular application" in claim 21 and the change management layer for performing that function in claim 1[3]. POR at 22-32; Ex. 2032 ¶¶51-84; Ex. 2033 ¶¶41-64; Ex. 1058 at 118:3-119:6, 127:5-128:8, 143:2-144:5 (Dr. Jagadish conceded that he offered no opinion that the prior art fails to meet any other claim limitation)[4].

Petitioner's Reply on '482 patent, Paper 72 at 4 in IPR2015-01751

For each of Grounds 1-3, the **only** limitation the POR and AIT's supporting expert declarations alleged was not met is the fourth portion configured to automatically detect "changes that affect the information in the first portion of the server or the information in the second portion of the server." POR at 22-32; Ex. 2032 ¶¶51-84; Ex. 2033 ¶¶41-64; Ex. 1058 at 118:3-119:6, 127:5-128:8, 143:2-144:5 (Dr. Jagadish conceded that he offered no opinion that the prior art fails to meet any other claim limitation)[3].

Petitioner's Reply on '111 patent, Paper 70 at 4-5 in IPR2015-01750

13. A system, comprising:

a server accessible by a browser executed on a client device, the server including a first portion, a second portion, a third portion, and a fourth portion,

the first portion of the server having information about unique aspects of a particular application,

the second portion of the server having information about user interface elements and one or more functions common to various applications, the various applications including the particular application,

the third portion of the server being configured to dynamically generate a functionality and a user interface for the particular application, the functionality and the user interface of the particular application being based on the information in the first portion of the server and the information in the second portion of the server, the third portion of the server being configured to send the functionality and the user interface for the particular application to the browser upon establishment of a connection between the server and the client device,

the fourth portion of the server being configured to automatically detect changes that affect the information in the first portion of the server or the information in the second portion of the server.

'111 patent, Ex. 1001 at 33:19 – 34:8 in IPR2015-01750

13. A system, comprising:

a server accessible by a browser executed on a client device, the server including a first portion, a second portion, a third portion, and a fourth portion,

the first portion of the server having information about unique aspects of a particular application,

the second portion of the server having information about user interface elements and one or more functions common to various applications, the various applications including the particular application,

the third portion of the server being configured to dynamically generate a functionality and a user interface for the particular application, th
interface of the particular
information in the first p
information in the second
portion of the server being
tionality and the user inte
cation to the browser upon
between the server and the client device,

the fourth portion of the server being configured to automatically detect changes that affect the information in the first portion of the server or the information in the second portion of the server.

the fourth portion of the server being configured to automatically detect changes that affect the information in the first portion of the server or the information in the second portion of the server.

'111 patent, Ex. 1001 at 33:19 – 34:8 in IPR2015-01750

23 specify -- **sounds to me like visiting different**
24 **websites, looking at them to see if changes have**
25 **taken place.**

Dr. Crovella indicates that the change management layer includes an agent that cruises the Web. The detected changes clearly "arise from changes external to the application."

D. The "automatically detecting" step

Method claim 21 of the '482 patent includes the step of "automatically detecting changes that affect an application" which corresponds to the "change management layer". The meaning of "automatically detecting" should correspond to that of the "change management later" and the "changes" therein should likewise "arise from changes external to the application."

E. "The fourth portion" or "the fourth portion of the server" element

Claim 13 of the '111 patent includes the "fourth portion" or the "fourth portion of the server". The full text of this limitation reads, "the fourth portion of the server being configured to automatically detect changes that affect the information in the first portion of the server or the information in the second portion of the server."

Substituting the "first portion" and the "second portion" limitations from claim 13 into the "fourth portion" limitation, we have:

> the fourth portion being configured to automatically detect changes that affect (i) information about unique aspects of a particular application, or (ii) information about user interface elements and one or more functions common to various applications including the particular application.

This limitation, especially the phrase, "changes that affect," is clearly the same as the "changes" discussed above regarding the change management layer. Thus, the "fourth portion," as understood by a person of ordinary skill in the art and like the "change management layer" automatically detects changes that "arise from changes external to the application."

F. The "intelligent agent" element

The '482 patent expressly defines "intelligent agent" in two places. First, the '482 patent states, "An 'intelligent agent' is a specialized program that makes decisions and performs tasks based on predefined rules and objectives." (20:1-3).

Second, the '482 patent states, "An 'intelligent agent' is a specialized program that resides on a network, or at a server as an applet, and can make decisions and perform tasks based on pre-defined rules." (10:42-45). This is consistent with the understanding of a person of ordinary skill in the art. (Ex. 2032, ¶¶ 49-50; Ex. 2033, ¶ 40). The second statement is somewhat narrower than the

POR, Paper No. 63 at 21 in IPR2015-01750

a change management layer for automatically detecting changes that affect an application,

'482 patent at 32:27-28, Ex. 1001 in IPR2015-01751, Ex. 1101 in IPR2015-01752

the fourth portion of the server being configured to automatically detect changes that affect the information in the first portion of the server or the information in the second portion of the server.

'111 patent at 34:5-8, Ex. 1001 in IPR2015-01750

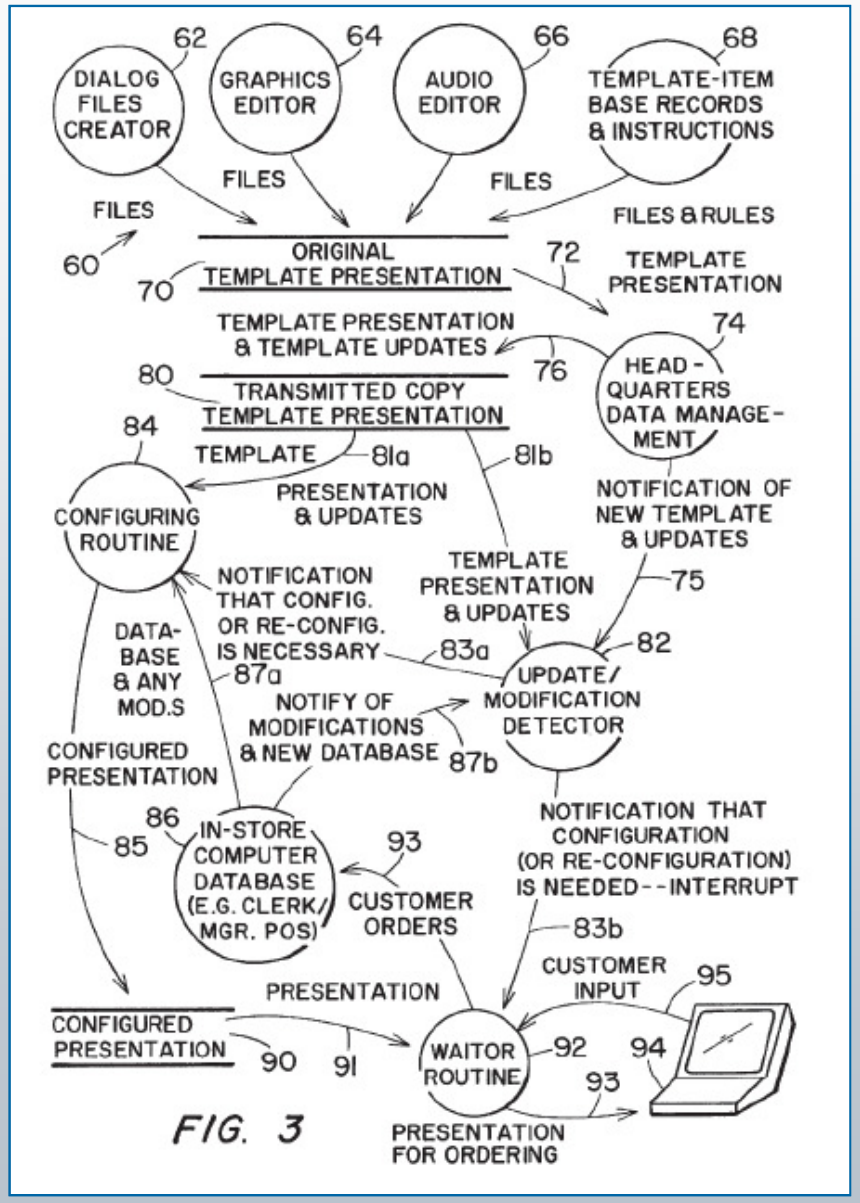


FIG. 3

Balderrama, FIG. 3, Ex. 1006/1106

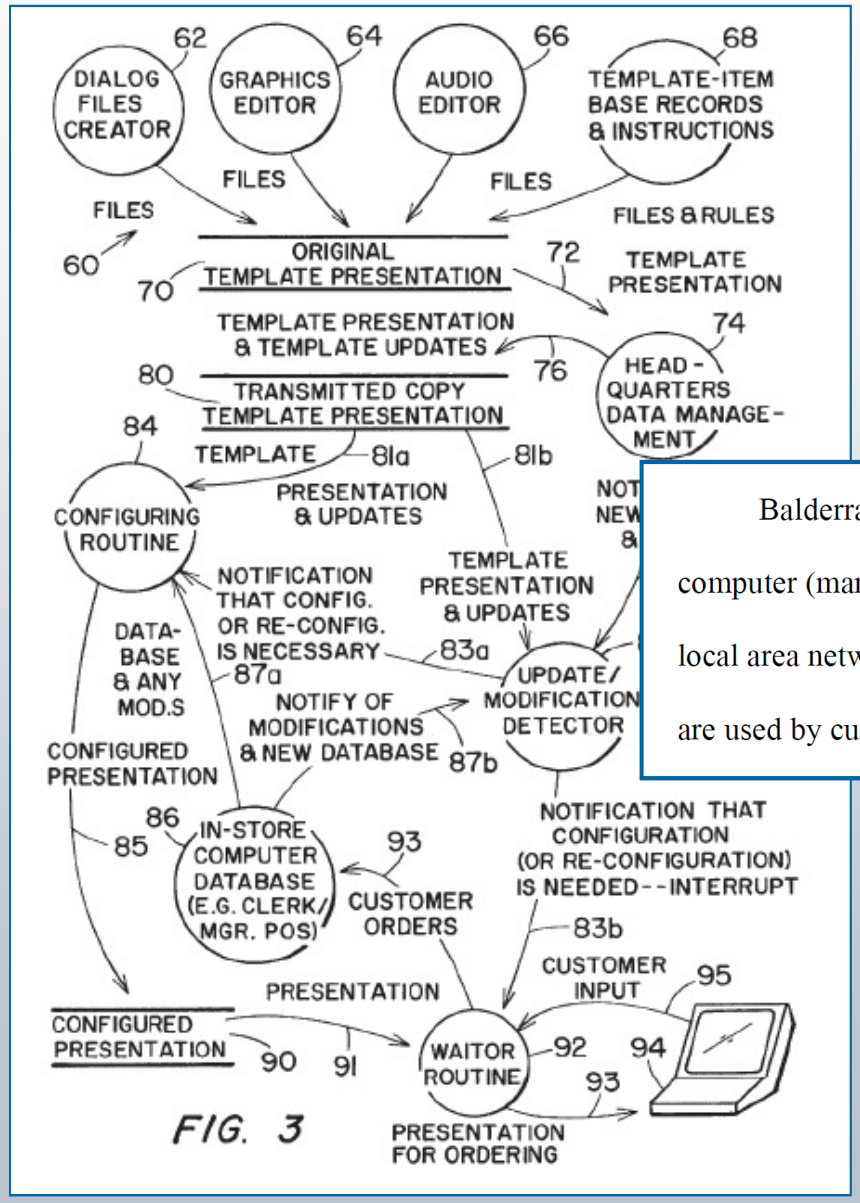


FIG. 3

Balderrama, FIG. 3, Ex. 1006/1106

Balderrama discloses a network system for a sales outlet, and employs a server computer (manager station 10) that distributes an order-entry presentation over a local area network (LAN) to client computers (customer terminals 20a, 20b, 20c) that are used by customers to enter orders. (FIG. 1; Crovella ¶¶ 145, 148-150; see also

Petition, Paper 1 at 34 in IPR2015-01750,
at 42 in IPR2015-01751,
at 25 in IPR2015-01752

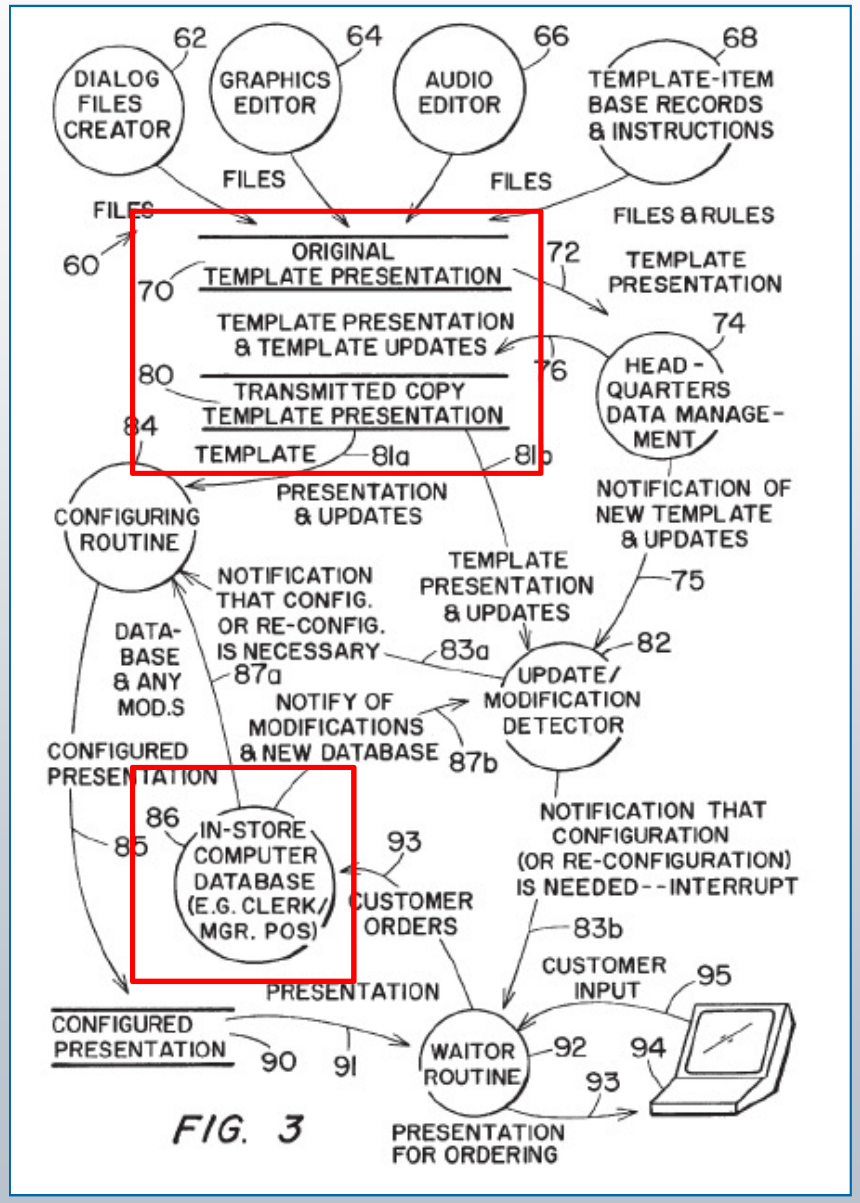


FIG. 3

Balderrama, FIG. 3, Ex. 1006/1106

At each sales outlet, the centrally developed order-entry UI is combined with outlet-specific data (e.g., the items for sale at a particular sales outlet, local prices for individual sales items) in a configuration process that tailors the presentation so as to be unique to the particular outlet. (1:15-23; Crovella ¶ 145.) The outlet-specific data files and records contain information about the unique aspects of a particular order-entry application and are stored in a database 86 located on, and therefore associated with, manager station 10 (the outlet's server). (9:16-27; Crovella ¶ 151.) Particular data files and records that are incorporated into the outlet's presentation (labeled with reference number 87a) correspond to the claimed "first layer associated with the server computer." (9:16-27; Crovella ¶ 151.)

The shared-across-outlets template presentation 80 from headquarters is transmitted to manager station 10 (the outlet's server) for combination with the outlet-specific data, and corresponds to the claimed "second layer associated with the server computer." (8:67-9:2; 11:43-46; Crovella ¶ 152; see also claim chart below.)

Petition, Paper 1 at 42-43 in IPR2015-01751, at 26 in IPR2015-01752

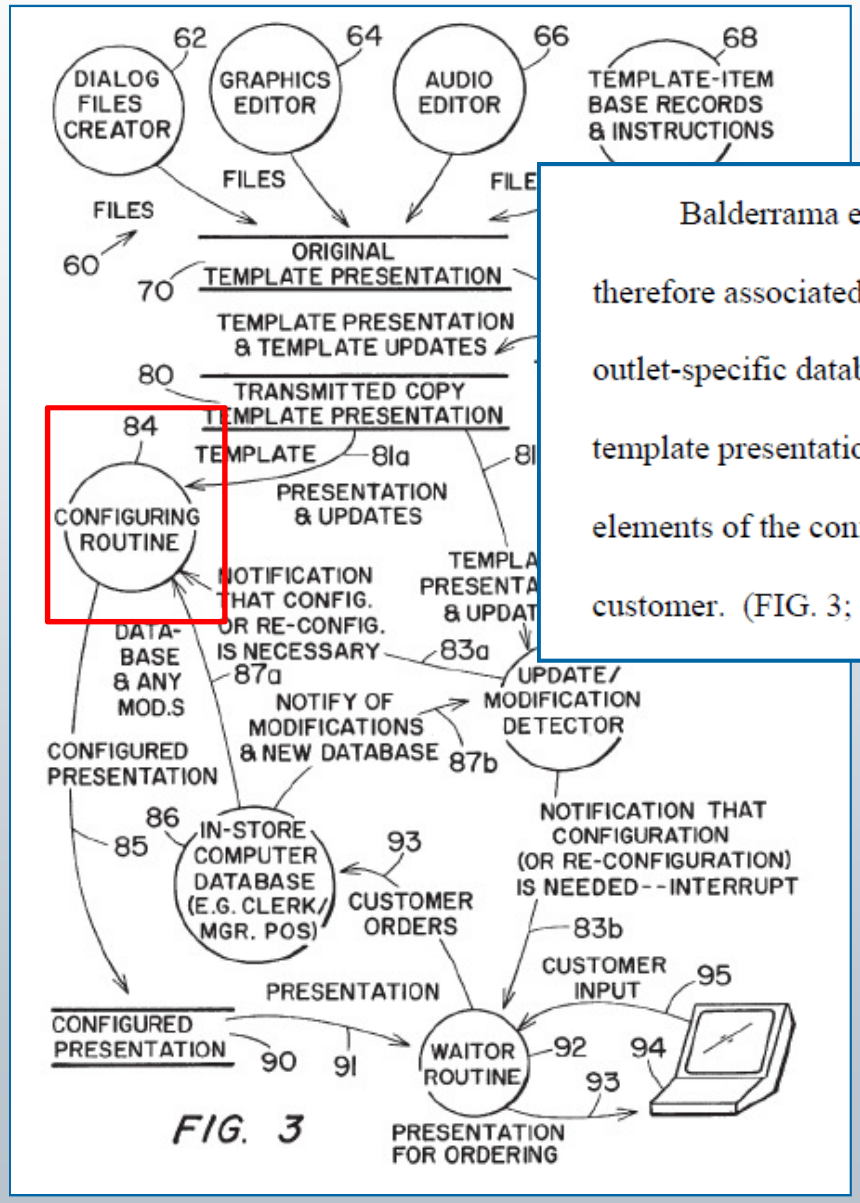
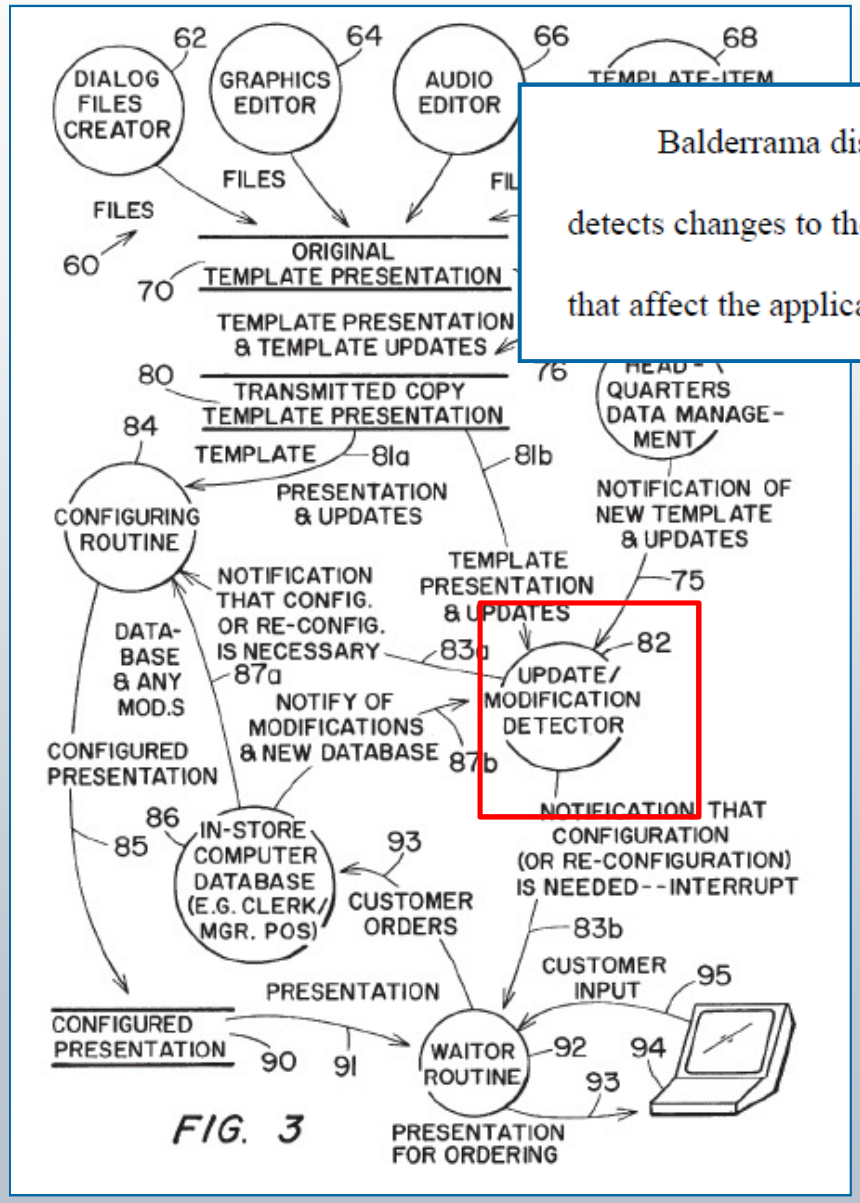


FIG. 3

Balderrama, FIG. 3, Ex. 1006/1106

Balderrama employs a configuring routine 84, which executes on, and is therefore associated with, manager station 10 (the server), to retrieve data from the outlet-specific database files/records (first layer) and combine it with the generic template presentation (second layer) in order to generate the functionality and UI elements of the configured presentation (application) for presentation to the customer. (FIG. 3; 11:38-46; Crovella ¶¶ 153-154.)

Petition, Paper 1 at 43 in IPR2015-01751, at 26-27 in IPR2015-01752



Balderrama discloses an update/modification detector 82, which automatically detects changes to the outlet-specific database or the generic template presentation that affect the application (the configured outlet-specific presentation). (10:14-21;

Petition, Paper 1 at 44 in IPR2015-01751, at 27 in IPR2015-01752

Balderrama, FIG. 3, Ex. 1006/1106

It would have been obvious to a POSA to implement a browser application on Balderrama's customer terminal for receiving and executing the order-entry application, as browsers (including Java-enabled browsers) were commonly used to receive UI applications in client-server systems as of the '482 patent's priority date. (Crovella ¶¶ 156-157.) For example, Java Complete (1996) describes using browsers for UI delivery over the Internet and within a company's internal network. (Ex. 1007 at 30, 31, 40; Crovella ¶ 156.)

Petition, Paper 1 at 45 in IPR2015-01751, at 28 in IPR2015-01752

A. Time period, yes.
Q. Would a thick client have been an obvious way to implement Balderrama?
A. For the cost reasons that I just mentioned, it would be an unnatural way to implement Balderrama.
Q. In the relevant time frame, had people used Java applets to deliver order entry systems to a computer running a browser?
A. So Java was relatively new. I think that by the late 1990s, which is what we're talking about, I believe Java applets would have been known and delivering functionality through Java applets to a browser would have been known. I'll need to verify to check the dates, but I think so. I think -- I think we're going to be good on that.
Q. And if the Balderrama system were implemented in that manner so that the presentation was downloaded to the local device as a Java applet and ran in a browser, would there be an application program running on the point of sale device?
A. Yeah. I believe that one could configure Balderrama or Balderrama with Java applet implementation to run an application at the point of sale device. I just believe that that is not the natural implementation that one would expect or that one would design as --

it's something that would feel inappropriate, not impossible, just inappropriate.
Q. So let's explore what you mean by unnatural and inappropriate. You're saying you don't think it would be the best way to do it?
A. I think that the combination of Java applets with Balderrama in the manner in which we just discussed would not be something that one of ordinary skill in the art would normally consider doing because it would be too expensive to be a useful practical system, so they probably wouldn't even go there.
Q. Well, I want to make sure I understand. So you think it might not be a commercially viable way of implementing it but concede it sounds that people would have understood it was an alternative? Is that a fair characterization of your testimony?
A. I think what I'm saying is that before you even get to putting two pieces together, you have to think about the two pieces in the same mind frame, and if the two pieces don't naturally go together because it would be expensive, for example, you aren't going to think about the two pieces in the same mind frame.
Q. So let me give you a hypothetical. If you were asked to offer an opinion on -- and Java applets for order presentation were known and Balderrama was known and I

Ex. 1058/1158 at 150:21-23

you put Balderrama and Java together, you could still be doing it as a thin client. So just because you threw Java into the mix doesn't necessarily say you're going to have a thick client or that you will have an applet.

Q. If the implementation we're talking about was the client runs the browser, functionality is downloaded as a Java applet, and I think you testified earlier that then there would be an application running on the client?

A. If that is the case, there would be an application running on the client.

Q. And your testimony about Balderrama not meeting the claim -- so when I asked whether it was based upon Balderrama being implemented with a thin client, you said no, so I guess I'd like to tease out when we walked through the limitations of the claim and I was talking about the presentation 90, you repeatedly said that's not an application program; right?

A. Correct.

Q. And that is based upon your assessment that it's running on a thin client; right?

A. I think if you combine Balderrama with an applet architecture, then you get something that is an applet encoding of the presentation, something like this. We

can choose what term to call it, be something like that. That thing we are agreeing would be an application. If you do not have such an applet encoding of the application or some other such means of putting what is currently Box 90 into an application without taking an extra step of somehow putting 90 into an application, if you just show me Balderrama and the Box 90, I don't see an application in Box 90.

Q. Well, we looked at the -- that introductory text in Column 1; right? Which says that presentation 90 is going to allow the sale of an item in a self-service fashion on an interactive device with a customer; right?

A. Yes.

Q. So to execute a sale, there's got to be an application program; right? Things have to be taken out of inventory, the finances. There needs to be some computer program running that is going to allow that point of sale device to actually complete a sale; right?

A. There needs to be some computer program running. That computer program may be running at the point of sale, may be running on the server, or on the combination thereof. Even if you did not have a thin client

Ex. 1058/1158 at 156:23 – 157:3

21. Thus, at a high level, the '482 and '111 patents describe metadata within two layers: one layer includes metadata that defines the unique aspects of an application; and the other layer includes metadata that defines aspects common to a variety of applications. They correspond to the first and second information, respectively, as recited in the asserted claims.

Ex. 1061/1161 at ¶ 21

Claim Term / Phrase	AIT Proposed Construction	Salesforce Proposed Construction
"changes that affect the information in the first portion of the server or the information in the second portion of the server" ('111 claim 13)	"changes to an application's metadata"	"modifications to regulatory, technological, or social requirements stored in a third party repository that affect information about unique aspects of a particular application or functions common to various applications"
"changes that affect a particular application"/ "changes that affect an application" ('482 claims 1, 21)	"changes to an application's metadata"	"modifications to regulatory, technological, or social requirements stored in a third party repository that affect an application"

Ex. 1060/1160 at 5

any of the claims of the '482 patent or the '111 patent. (Ex. 2032, ¶ 71-72; Ex. 2033, ¶ 55).

All of the "changes" in Kovacevic are accepted as a part of the application – namely user interaction accepted by the UI – which are not "external to the application." (Ex. 2032, ¶ 71; Ex. 2033, ¶ 54). Therefore, under the broadest reasonable interpretation in light of the specification, it is clear that Kovacevic does not anticipate claim 1 of the '482 patent. (Ex. 2032, ¶¶ 71-72; Ex. 2033, ¶ 55). Similarly, claim 21, which involves a method for automatically detecting "changes" that affect a particular application, cannot be anticipated by Kovacevic. (Ex. 2032, ¶ 72; Ex. 2033, ¶ 55). The remaining dependent claims are not anticipated by Kovacevic by virtue of their dependencies on claims 1 and 21. Likewise, claims 13-18 of the '111 patent cannot be anticipated by Kovacevic because Kovacevic does not disclose the required "fourth portion." (Ex. 2032, ¶ 73; Ex. 2033, ¶ 55).

C. Balderrama does not disclose a "change management layer", "automatically detecting a change" or a "fourth portion" and therefore cannot render any claims of the '482 patent or the '111 patent obvious

Balderrama generally describes a system which enables a series of point of sale ("POS") systems to be updated with pricing changes, item changes and the

like from a remote location by a manager. (Ex. 2032, ¶¶ 74-75; Ex. 2033, ¶ 56). To do so, Balderrama relies upon "templates". (*see, e.g.* Balderrama Col. 2, lines 10-21). The primary example given in Balderrama is a taco restaurant with updates to the menuin the POS. (*see, e.g.* Balderrama at Col. 7, TABLE A). The Petitioner did not rely upon Java Complete as disclosing a "change management layer" and instead pointed to Balderrama at Col. 10, lines 14-21 as disclosing this limitation.

Petitioner, and in turn the Board, relied upon an "update/modification detector 82" (e.g. Balderrama Col. 10, lines 10-24; 11, lines 64-67) as disclosing the "change management layer" (Order at 41). However, just as with Popp and Kovasevic, a user manually inputs new data into a database that is then incorporated into the system. (Ex. 2032, ¶ 777; Ex. 2033, ¶ 56) (*see also* Balderrama at Col. 10, lines 6-9). In Balderrama that user is "most-likely [a] manager[]". (*Id.*). The "update/modification detector 82" of Balderrama essentially detects user input, just like Popp and Kovacevic. One of ordinary skill in the art would not recognize the "update/modification detector 82" as "change management layer" that detects "changes" under the broadest reasonable interpretation.

Petitioner pointed to, and the Board relied upon Balderrama at 10:14-21, which describes the "update/modification Detector 82" as able to "take appropriate

POR at 28-29, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751

a change management layer for automatically detecting changes that affect an application,

'482 patent at 32:27-28 (claim 1), Ex. 1001 in IPR2015-01751, Ex. 1101 in IPR2015-01752

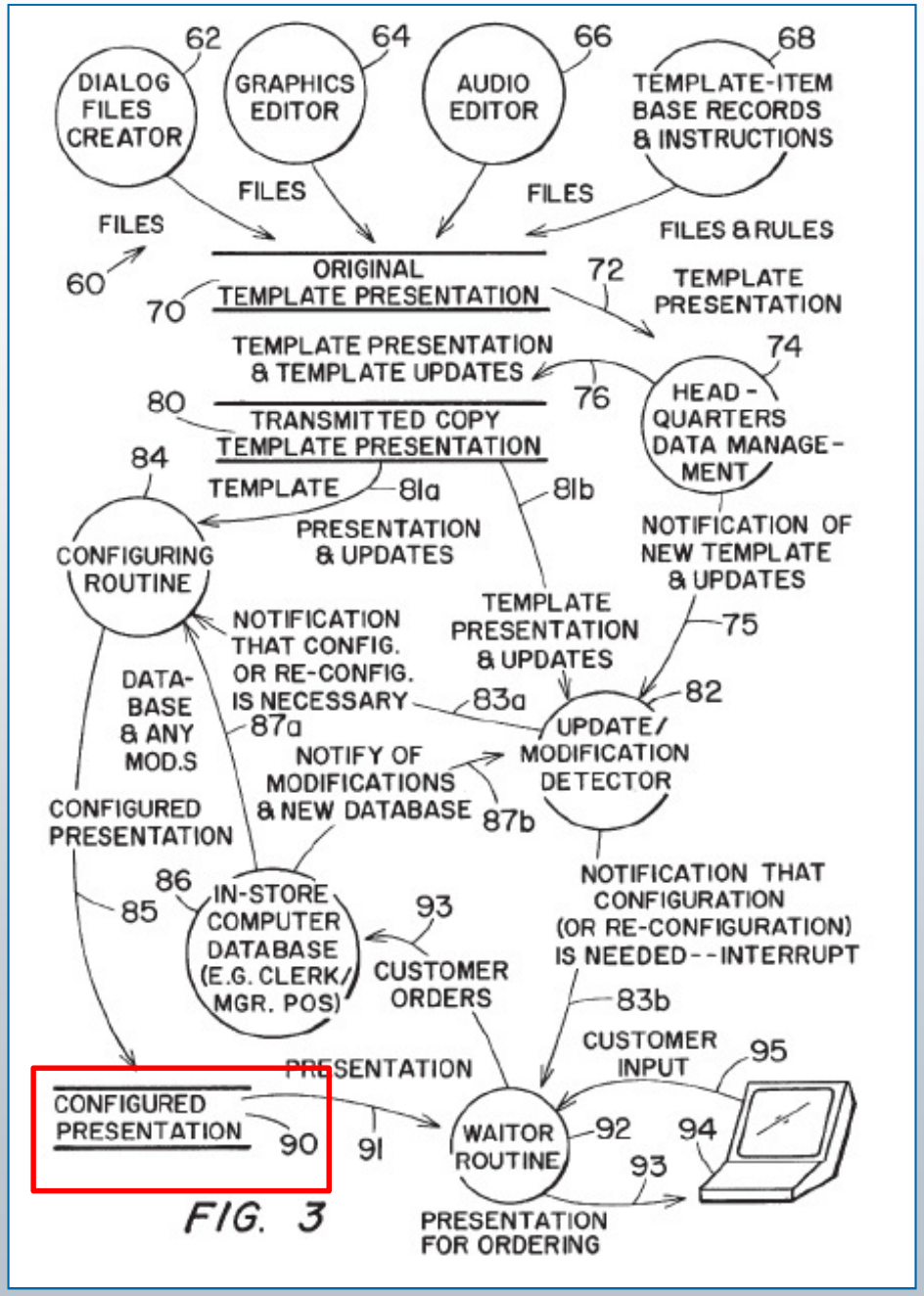


FIG. 3

RPX Expert on Balderrama, FIG. 3, Ex. 1006/1106

14. I disagree with Dr. Jagadish's assertion in ¶ 78 of his declaration that the changes detected in Balderrama "relate to a user performing an internal change to the application." The changes detected by Balderrama's update/modification detector 82 include modifications to files and records in database 86 (corresponding to the claimed "first layer" or "first portion of the server") and updates to template presentation 80 (corresponding to the claimed "second layer" or "second portion of the server"); whereas the configured presentation 90 corresponds to the claimed "application." When a user at a sales outlet makes a modification to files and records in database 86 (Ex. 1006 at 10:7-10), or when another user (e.g., at corporate headquarters) makes updates to template presentation 80 (*Id.* at 8:16-67), those changes are not input directly to configured presentation 90. Configured presentation 90 is the application that results from subsequently bringing together the changed database files and records (first layer/portion) and template presentation (second layer/portion). A POSA would have understood that the upstream changes made to those separate (first and second) layers/portions are in fact external to the application (configured presentation 90), in addition to "arising from" changes external to the application (see ¶ 12 above). See, e.g., Ex. 1006 at 8:16-64, discussing identifying updates to template presentation at corporate headquarters, or at the facility of a third party handling software service/support, etc., which a POSA would have understood to be external to the application.

Ex. 1062/1162 at ¶ 14

Balderrama discloses an update/modification detector 82, which automatically detects changes to the outlet-specific database or the generic template presentation that affect the application (the configured outlet-specific presentation). (10:14-21;

Petition, Paper 1 at 44 in IPR2015-01751, at 27 in IPR2015-01752

The flow diagram in FIG. **4** represents one pr process **100** of the invention. After an original te presentation has been created **102***b* and a database c records provided **102***a*, configuration **104** of ele information for presentation at an interactive electronic device takes place utilizing a configuring routine (as represented in FIG. **3** at **84**). Configuration preferably takes place, to increase efficiency and outlet flexibility, on a computer processor located at the sales outlet (such as the manager's station **10**, one or all of the clerk POS terminals **12***a*, **12***b*, **12***c*, or one or all of the customer terminals **20***a*, **20***b*, **20***c*—all shown in FIG. **1**). However, this is not necessary. Original template presentations may be created and configured with local databases at the same location, or a template and database could be configured at a third party software service/support facility, and then transmitted to the store (with a communication system as shown in FIG. **2**) for downloading to a customer terminal. It is important that the new process and system allow for database modifications **106** (described above in connection with FIG. **3**), as well as original template presentation updates **108** (such as: providing a new template presentation; modifying instructions or dialog control files within the original template; making the presentation of an item time-dependent over some period of time (such as certain hours in a day, or days in a week); adding or deleting a data record associated with an item; and modifying a field containing an icon, motiongraphics/animation message, a video or audio message, etc.). If database modifications or template updates are made (**110**, **114**), re-configuration **116** is necessary to change the presentation to a customer. If no modifications or updates have been made (**110**, **112**), no subsequent configuration, or re-configuration, would be necessary **118**. A loop **119** has been formed to represent that, as time passes, the database may need to be modified and/or the original template may need updating.

Balderrama, Ex. 1006/1106 at 11:64-67

12. The Balderrama/Java Complete combination meets the claim limitations referenced in ¶ 7 above even when construed using Dr. Jagadish's overly narrow construction. Balderrama's update/modification detector 82 automatically detects changes including modifications to files and records in database 86 and updates to template presentation 80. (Ex. 1006[4] at FIG. 3; 2:16-21; 10:14-21; 11:64-67; 12:34-38.) These changes impact how the application (configured presentation 90) should operate, because they trigger re-configuration of presentation 90 to present updated layouts, icons, graphics, items for sale, prices, specials, branch cells, etc. (*Id.* at 6:48-63; 10:11-24.) For example, a change to a branch cell in template presentation 80 would change whether the configured presentation 90 is to call up a particular screen at a particular point in its execution, thus impacting how the application should operate. (*Id.* at 6:51-55.) The detected changes to the files and records in database 86 arise from changes external to the application program, such as a change in the set of items that a particular sales outlet offers for sale, a change in the sales outlet's price of an item for sale, etc. (*Id.* at 10:14-21.) The detected changes to the template presentation 80 also arise from changes external to the application program, such as a change in how corporate headquarters personnel require the presentation of items for sale to be laid out, a change in the graphics and messages that corporate headquarters chooses to include in all sales outlets' presentations, etc. (*Id.* at 6:48-63.) Update/modification detector 82 is labeled and described as a "detector;" it detects when updates to the template presentation and modifications to the database files and records necessitate reconfiguration of the presentation 90, for example by detecting which portions of the presentation are affected by the modifications or updates and therefore require reconfiguration. (*Id.* at 12:39-44.) This detection is performed automatically, without human involvement in the detection. Update/modification detector 82 thus automatically detects changes that arise from changes external to the application program, and the automatically detected changes impact how the application program should operate.

RPX's Expert, Ex. 1062/1162 at ¶ 12

1. A system for providing a dynamically generated application having one or more functions and one or more user interface elements; comprising:

a server computer;

one or more client computers connected to the server computer over a computer network;

a first layer associated with the server computer containing information about the unique aspects of a particular application;

a second layer associated with the server computer containing information about the user interface and functions common to a variety of applications, a particular application being generated based on the data in both the first and second layers;

a third layer associated with the server computer that retrieves the data in the first and second layers in order to generate the functionality and user interface elements of the application; and

a change management layer for automatically detecting changes that affect an application,

each client computer further comprising a browser application being executed by each client computer, wherein a user interface and functionality for the particular application is distributed to the browser application and dynamically generated when the client computer connects to the server computer.

'482 Patent (Ex. 1001), claim 1.

a change management layer for automatically detecting changes that affect an application,

27. In my opinion, the broadest reasonable interpretation a POSITA would apply to a "change management layer" is automatically detecting changes which impact how the application program should operate. In the context of the '482 patent, these "changes" detected by the change management layer arise from changes external to the application program.

RPX Exhibit 1057
RPX v. AIT
IPR2015-01751

RPX Exhibit 1057
RPX v. AIT

Jagadish Decl. (Ex. 2032) at ¶ 27.

12. The Balderrama/Java Complete combination meets the claim limitations referenced in ¶ 7 above even when construed using Dr. Jagadish's overly narrow construction. Balderrama's update/modification detector 82 automatically detects changes including modifications to files and records in database 86 and updates to template presentation 80. (Ex. 1006[4] at FIG. 3; 2:16-21; 10:14-21; 11:64-67; 12:34-38.) These changes impact how the application (configured presentation 90) should operate, because they trigger re-configuration of presentation 90 to present updated layouts, icons, graphics, items for sale, prices, specials, branch cells, etc. (*Id.* at 6:48-63; 10:11-24.) For example, a change to a branch cell in template presentation 80 would change whether the configured presentation 90 is to call up a particular screen at a particular point in its execution, thus impacting how the application should operate. (*Id.* at 6:51-55.) The detected changes to the files and records in database 86 arise from changes external to the application program, such as a change in the set of items that a particular sales outlet offers for sale, a change in the sales outlet's price of an item for sale, etc. (*Id.* at 10:14-21.) The detected changes to the template presentation 80 also arise from changes external to the application program, such as a change in how corporate headquarters personnel require the presentation of items for sale to be laid out, a change in the graphics and messages that corporate headquarters chooses to include in all sales outlets' presentations, etc. (*Id.* at 6:48-63.) Update/modification detector 82 is labeled and described as a "detector;" it detects when updates to the template presentation and modifications to the database files and records necessitate reconfiguration of the presentation 90, for example by detecting which portions of the presentation are affected by the modifications or updates and therefore require reconfiguration. (*Id.* at 12:39-44.) This detection is performed automatically, without human involvement in the detection. Update/modification detector 82 thus automatically detects changes that arise from changes external to the application program, and the automatically detected changes impact how the application program should operate.

RPX's Expert, Ex. 1062/1162 at ¶ 12

delaying when a special is made available; right?
A. Did you say Column 10 or some other --
Q. Yeah. 10, Lines 10 to 25.
A. But that's where we are right now.
Q. Yeah. If you just keep reading.
A. Yes.
Q. So a price change is an example of the type of change that the update detector 82 detects; right?
A. Yes. There is an update/modification detector module 82 that is shown in Figure 3 of Balderrama and described -- at least partly described in Column 10 that we are talking about, and this module is responsible for taking appropriate action when there are database or template presentation updates.
Q. And in the last sentence in Paragraph 75 of your declaration you say that the update/modification detector 82 detects when a manager has updated a database in a way that will impact the menu on the point of sale device; right?
A. That is correct.
Q. And so a price change, for example, of an item would be an example of the type of database change that you're referring to that the update/modification detector 82 will detect; right?
A. That is correct.

Q. And when the update/modification detector 82 detects this change to the database, the result will be that the menu -- or the menu or presentation 90 on the point of sale display will be updated, for example, to reflect the new price for the item based upon the change in the database; is that right?
A. Right. The way this system works is when there is say a price change for a particular item, this -- the new price is input by a manager, by a user into the database, and whichever menus include this item where this price has been let's say increased will have the -- the menu presentation is updated to reflect the new price, so the manager has to input this thing once the change to the price is input once by -- as user input from the manager, and it is then propagated by the Balderrama system to in general multiple presentations that are impacted.
Q. At the point of sale devices?
A. At the point of sale devices.
Q. Right. And in the second sentence of Paragraph 74 of your declaration you state that Balderrama provides functionality and associated systems that enable a manager to update that menu, I'm skipping a little bit, and have those changes reflected on each of the associated points of sale.

AIT's Expert, Ex. 1058/1158 at 132:21 – 133:7

12. The Balderrama/Java Complete combination meets the claim limitations referenced in ¶ 7 above even when construed using Dr. Jagadish's overly narrow construction. Balderrama's update/modification detector 82 automatically detects changes including modifications to files and records in database 86 and updates to template presentation 80. (Ex. 1006[4] at FIG. 3; 2:16-21; 10:14-21; 11:64-67; 12:34-38.) These changes impact how the application (configured presentation 90) should operate, because they trigger re-configuration of presentation 90 to present updated layouts, icons, graphics, items for sale, prices, specials, branch cells, etc. (*Id.* at 6:48-63; 10:11-24.) For example, a change to a branch cell in template presentation 80 would change whether the configured presentation 90 is to call up a particular screen at a particular point in its execution, thus impacting how the application should operate. (*Id.* at 6:51-55.) The detected changes to the files and records in database 86 arise from changes external to the application program, such as a change in the set of items that a particular sales outlet offers for sale, a change in the sales outlet's price of an item for sale, etc. (*Id.* at 10:14-21.) The detected changes to the template presentation 80 also arise from changes external to the application program, such as a change in how corporate headquarters personnel require the presentation of items for sale to be laid out, a change in the graphics and messages that corporate headquarters chooses to include in all sales outlets' presentations, etc. (*Id.* at 6:48-63.) Update/modification detector 82 is labeled and described as a "detector;" it detects when updates to the template presentation and modifications to the database files and records necessitate reconfiguration of the presentation 90, for example by detecting which portions of the presentation are affected by the modifications or updates and therefore require reconfiguration. (*Id.* at 12:39-44.) This detection is performed automatically, without human involvement in the detection. Update/modification detector 82 thus automatically detects changes that arise from changes external to the application program, and the automatically detected changes impact how the application program should operate.

RPX's Expert, Ex. 1062/1162 at ¶ 12

12. The Balderrama/Java Complete combination meets the claim limitations referenced in ¶ 7 above even when construed using Dr. Jagadish's overly narrow construction. Balderrama's update/modification detector 82 automatically detects changes including modifications to files and records in database 86 and updates to template presentation 80. (Ex. 1006[4] at FIG. 3; 2:16-21; 10:14-21; 11:64-67; 12:34-38.) These changes impact how the application (configured presentation 90) should operate, because they trigger re-configuration of presentation 90 to present updated layouts, icons, graphics, items for sale, prices, specials, branch cells, etc. (*Id.* at 6:48-63; 10:11-24.) For example, a change to a branch cell in template presentation 80 would change whether the configured presentation 90 is to call up a particular screen at a particular point in its execution, thus impacting how the application should operate. (*Id.* at 6:51-55.) The detected changes to the files and records in database 86 arise from changes external to the application program, such as a change in the set of items that a particular sales outlet offers for sale, a change in the sales outlet's price of an item for sale, etc. (*Id.* at 10:14-21.) The detected changes to the template presentation 80 also arise from changes external to the application program, such as a change in how corporate headquarters personnel require the presentation of items for sale to be laid out, a change in the graphics and messages that corporate headquarters chooses to include in all sales outlets' presentations, etc. (*Id.* at 6:48-63.) Update/modification detector 82 is labeled and described as a "detector;" it detects when updates to the template presentation and modifications to the database files and records necessitate reconfiguration of the presentation 90, for example by detecting which portions of the presentation are affected by the modifications or updates and therefore require reconfiguration. (*Id.* at 12:39-44.) This detection is performed automatically, without human involvement in the detection. Update/modification detector 82 thus automatically detects changes that arise from changes external to the application program, and the automatically detected changes impact how the application program should operate.

RPX's Expert, Ex. 1062/1162 at ¶ 12

12. The Balderrama/Java Complete combination meets the claim limitations referenced in ¶ 7 above even when construed using Dr. Jagadish's overly narrow construction. Balderrama's update/modification detector 82 automatically detects changes including modifications to files and records in database 86 and updates to template presentation 80. (Ex. 1006[4] at FIG. 3; 2:16-21; 10:14-21; 11:64-67; 12:34-38.) These changes impact how the application (configured presentation 90) should operate, because they trigger re-configuration of presentation 90 to present updated layouts, icons, graphics, items for sale, prices, specials, branch cells, etc. (*Id.* at 6:48-63; 10:11-24.) For example, a change to a branch cell in template presentation 80 would change whether the configured presentation 90 is to call up a particular screen at a particular point in its execution, thus impacting how the application should operate. (*Id.* at 6:51-55.) The detected changes to the files and records in database 86 arise from changes external to the application program, such as a change in the set of items that a particular sales outlet offers for sale, a change in the sales outlet's price of an item for sale, etc. (*Id.* at 10:14-21.) The detected changes to the template presentation 80 also arise from changes external to the application program, such as a change in how corporate headquarters personnel require the presentation of items for sale to be laid out, a change in the graphics and messages that corporate headquarters chooses to include in all sales outlets' presentations, etc. (*Id.* at 6:48-63.) Update/modification detector 82 is labeled and described as a "detector;" it detects when updates to the template presentation and modifications to the database files and records necessitate reconfiguration of the presentation 90, for example by detecting which portions of the presentation are affected by the modifications or updates and therefore require reconfiguration. (*Id.* at 12:39-44.) This detection is performed automatically, without human involvement in the detection. Update/modification detector 82 thus automatically detects changes that arise from changes external to the application program, and the automatically detected changes impact how the application program should operate.

RPX's Expert, Ex. 1062/1162 at ¶ 12

Accordingly, the "change management layer" and the associated "changes" are not shown in the combination of Balderrama and Java Complete. When those terms are read under the broadest reasonable interpretation in light of the specification, claim 1 of the '482 patent cannot be obvious. (Ex. 2032, ¶ 81; Ex. 2033, ¶ 62). Similarly, claim 21, including a method to automatically detect "changes" that affect a particular application, cannot be obvious in view of Balderrama and Java Complete. (*Id.*). The remaining dependent claims are not anticipated by Balderrama in view of Java Complete by virtue of their dependencies on claims 1 and 21. Similarly, the Balderrama in view of Java Complete cannot render any claim of the "fourth portion" of claim 13 of the '111 patent obvious. (Ex. 2032, ¶ 82; Ex. 2033, ¶ 62).

POR, Paper 63 at 32 in IPR2015-01750

the fourth portion of the server being configured to automatically detect changes that affect the information in the first portion of the server or the information in the second portion of the server.

'111 patent, Ex. 1001 at 34:5-8 in IPR2015-01750

14. I disagree with Dr. Jagadish's assertion in ¶ 78 of his declaration that the changes detected in Balderrama "relate to a user performing an internal change to the application." The changes detected by Balderrama's update/modification detector 82 include modifications to files and records in database 86 (corresponding to the claimed "first layer" or "first portion of the server") and updates to template presentation 80 (corresponding to the claimed "second layer" or "second portion of the server"); whereas the configured presentation 90 corresponds to the claimed "application." When a user at a sales outlet makes a modification to files and records in database 86 (Ex. 1006 at 10:7-10), or when another user (e.g., at corporate headquarters) makes updates to template presentation 80 (*Id.* at 8:16-67), those changes are not input directly to configured presentation 90. Configured presentation 90 is the application that results from subsequently bringing together the changed database files and records (first layer/portion) and template presentation (second layer/portion). A POSA would have understood that the upstream changes made to those separate (first and second) layers/portions are in fact external to the application (configured presentation 90), in addition to "arising from" changes external to the application (see ¶ 12 above). See, e.g., Ex. 1006 at 8:16-64, discussing identifying updates to template presentation at corporate headquarters, or at the facility of a third party handling software service/support, etc., which a POSA would have understood to be external to the application.

RPX's Expert, Ex. 1062/1162 at ¶ 14

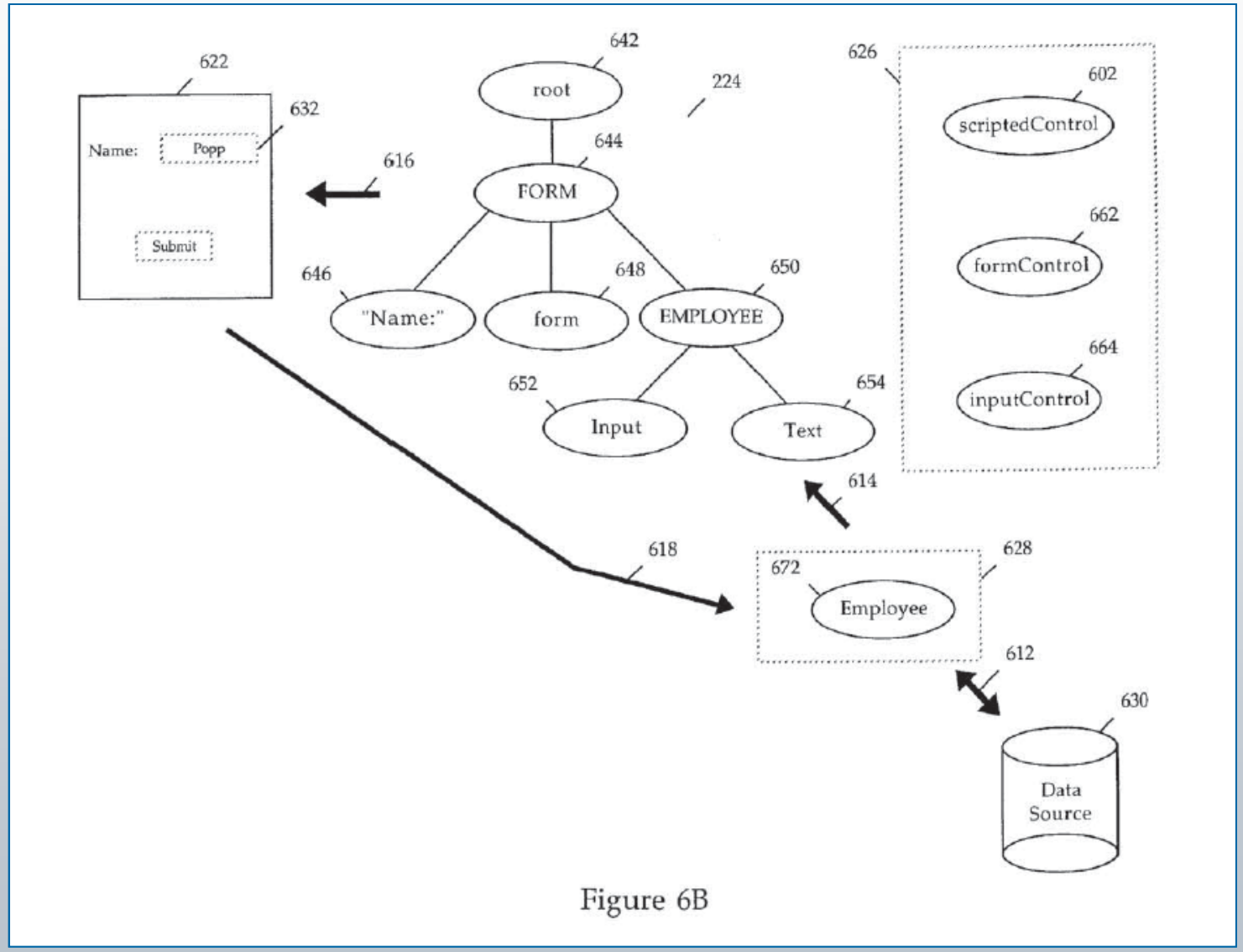
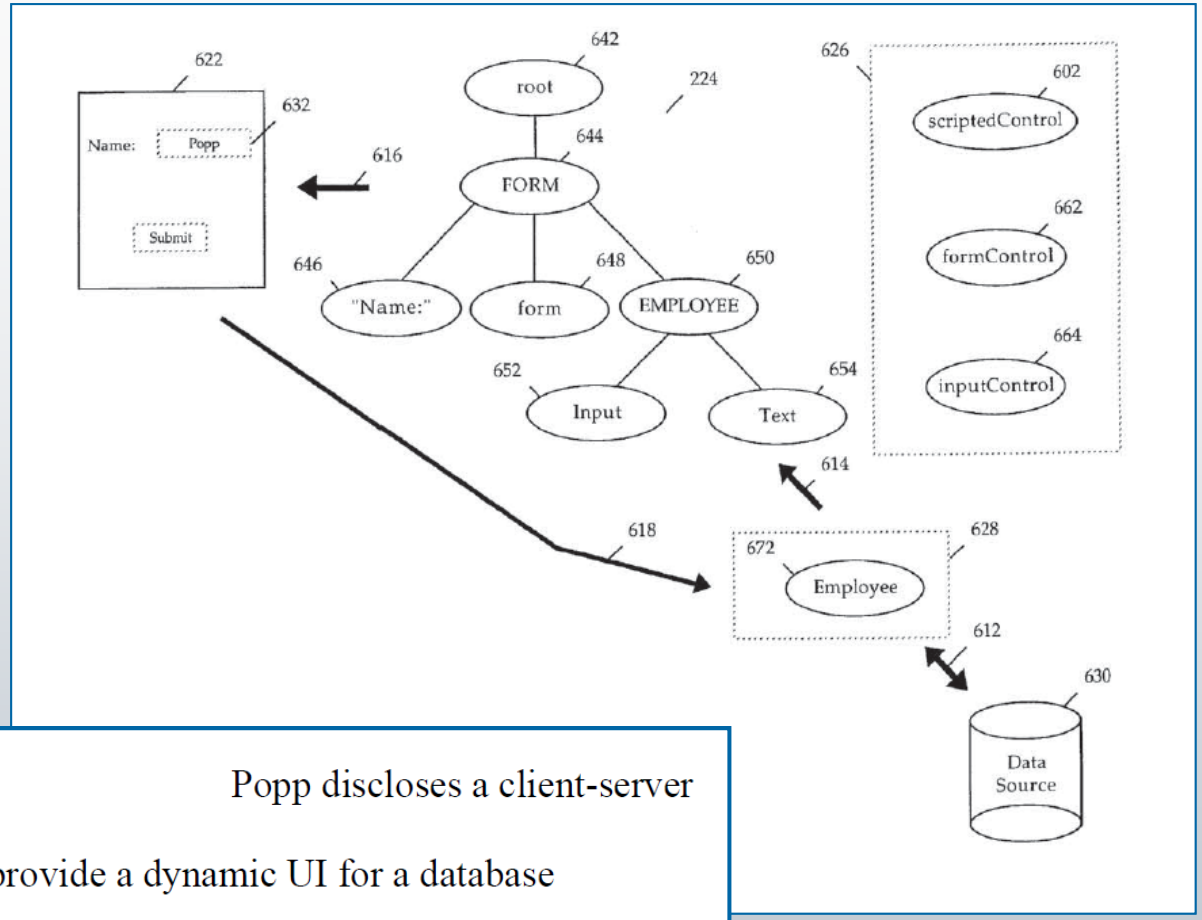


Figure 6B

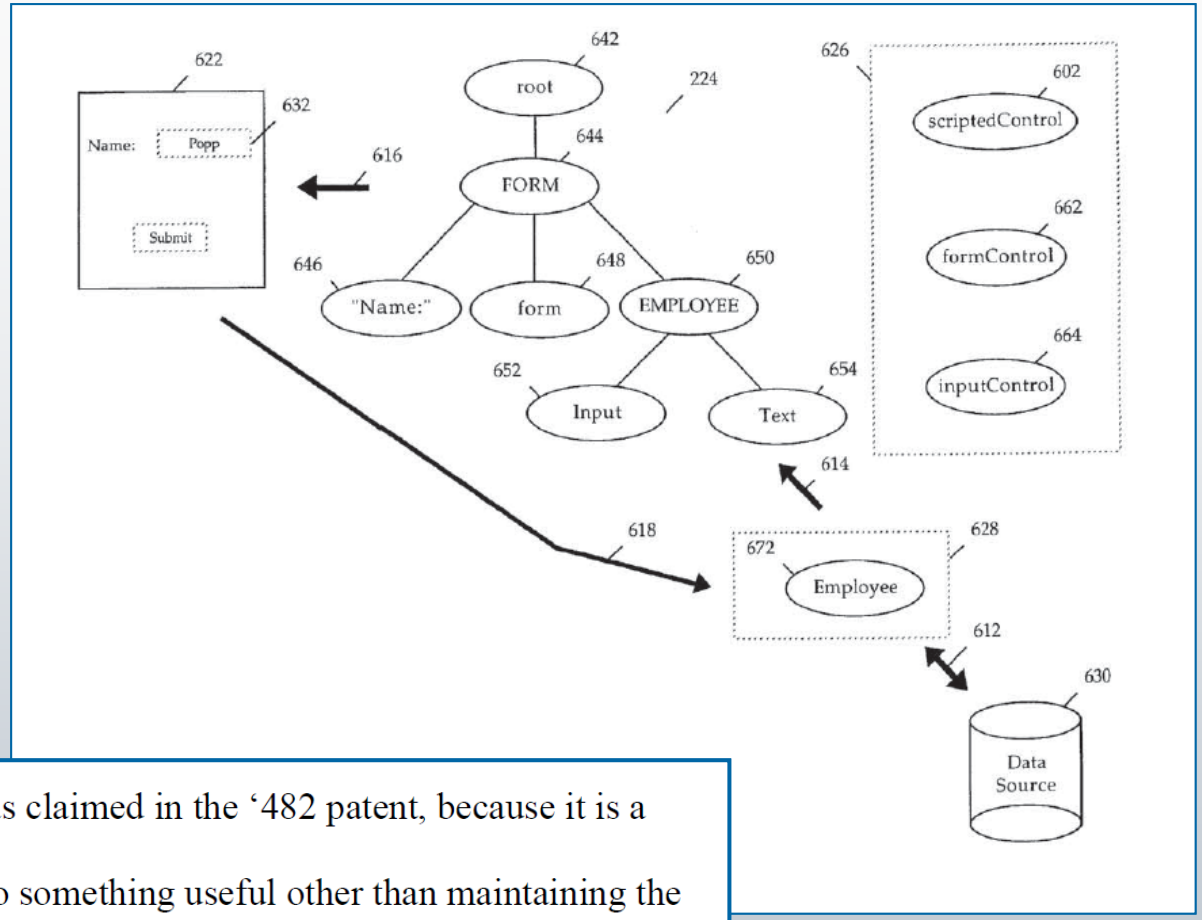
Popp, FIG. 6B, Ex. 1004/1104



Popp discloses a client-server system for generating Web pages that provide a dynamic UI for a database application that can respond to user input. (3:61-65; 8:24-26; Crovella ¶¶ 29-35.)

Popp, FIG. 6B, Ex. 1004/1104

Petition, Paper 1 at 16 in IPR2015-01751, at 15 in IPR2015-01752



Popp, FIG. 6B, Ex. 1004/1104

A Web page is an "application" as claimed in the '482 patent, because it is a program executable by a computer to do something useful other than maintaining the computer itself, e.g., displaying information to a user, eliciting and receiving input from the user, etc. (Crovella ¶ 31; see § VI.A *supra*). The '482 patent specification

Petition, Paper 1 at 16 in IPR2015-01751, at 15 in IPR2015-01752

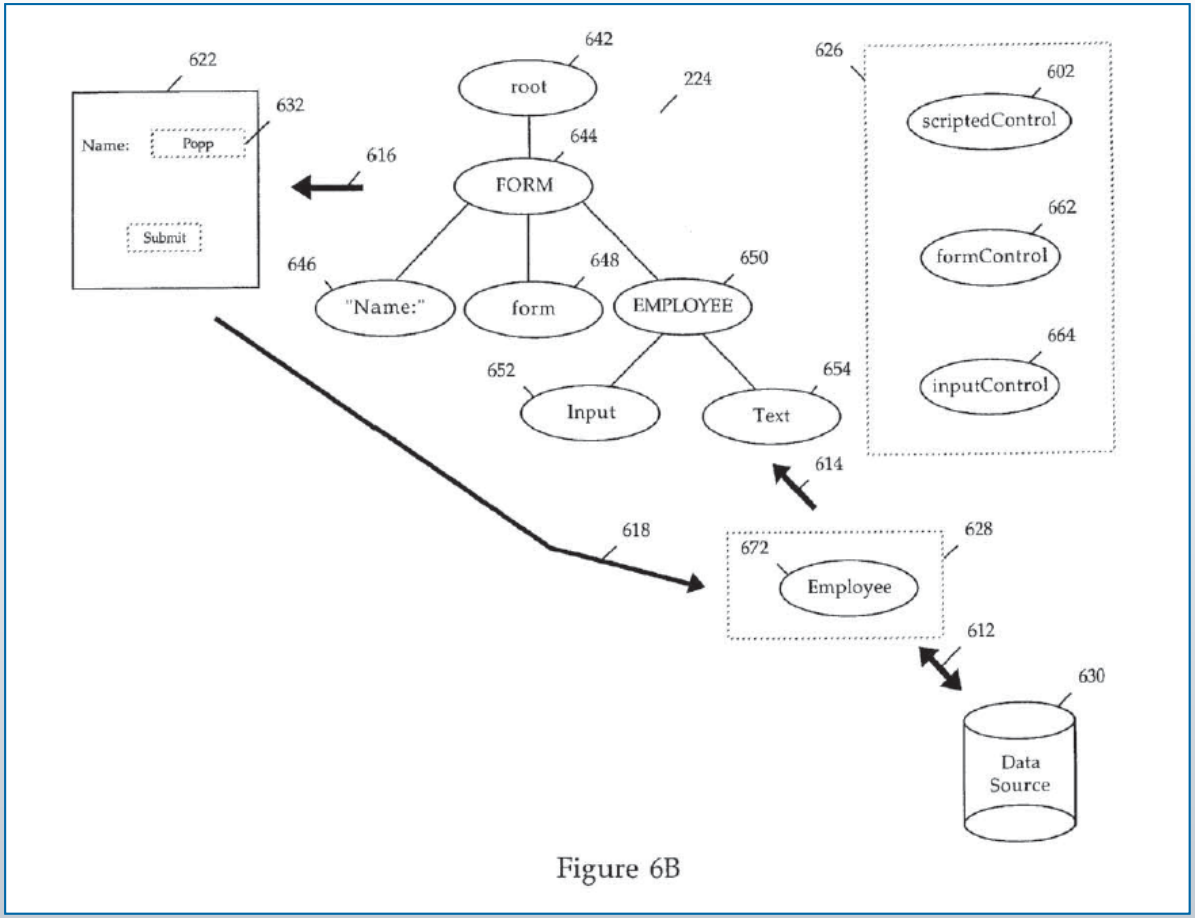


Figure 6B

Popp, FIG. 6B, Ex. 1004/1104

Popp's system separates application-specific data from application-generic presentation (e.g., UI) components,

Petition, Paper 1 at 17 in IPR2015-01751, at 16 in IPR2015-01752

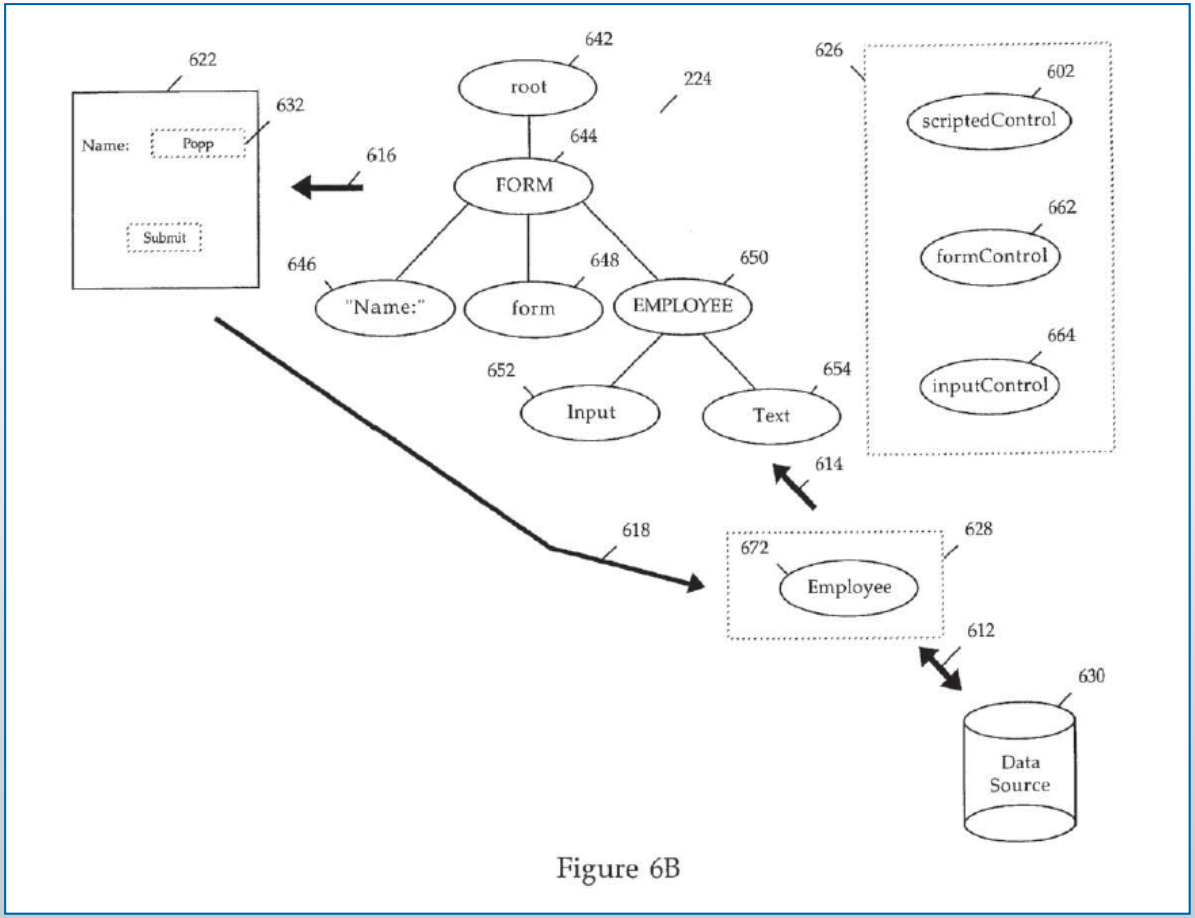
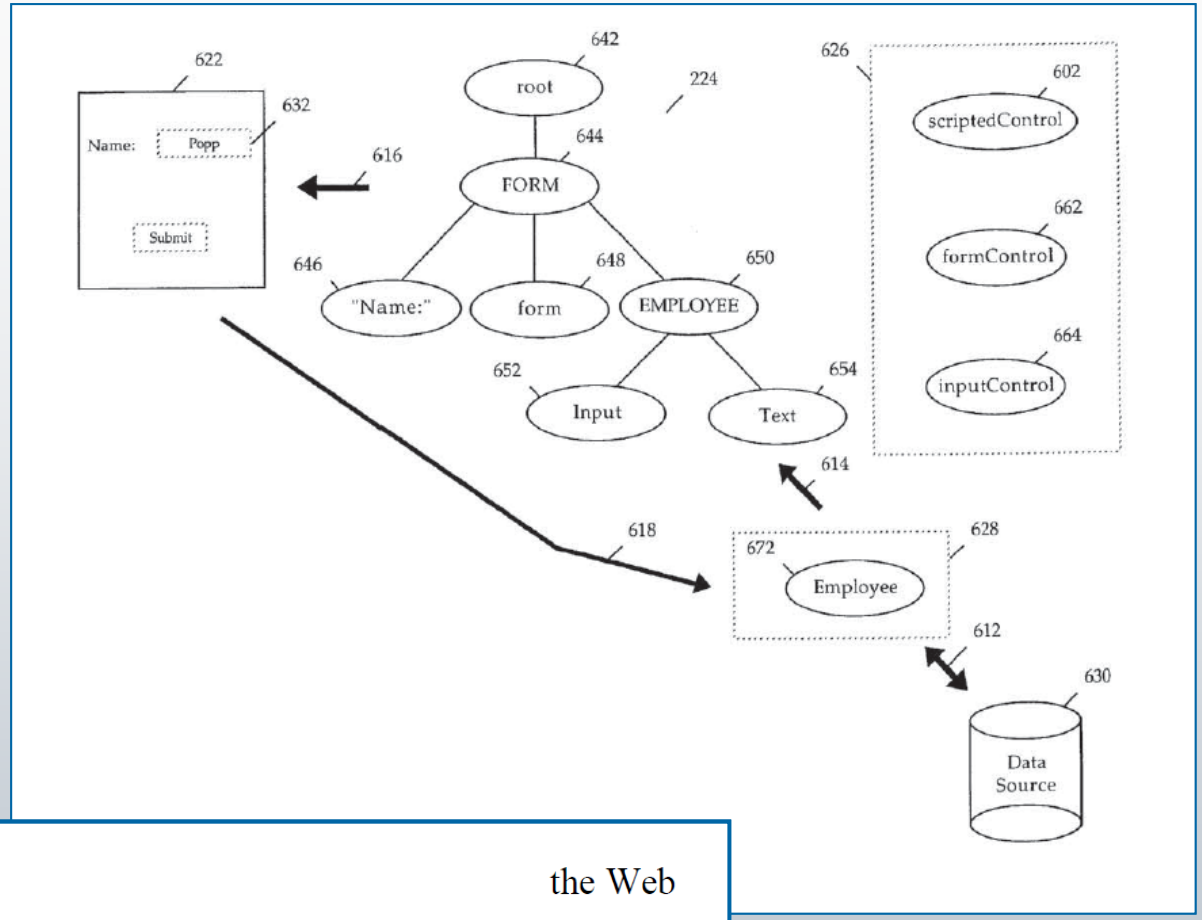


Figure 6B

Popp, FIG. 6B, Ex. 1004/1104

The database containing application-specific data corresponds to the "first layer" claimed in the '482 patent,

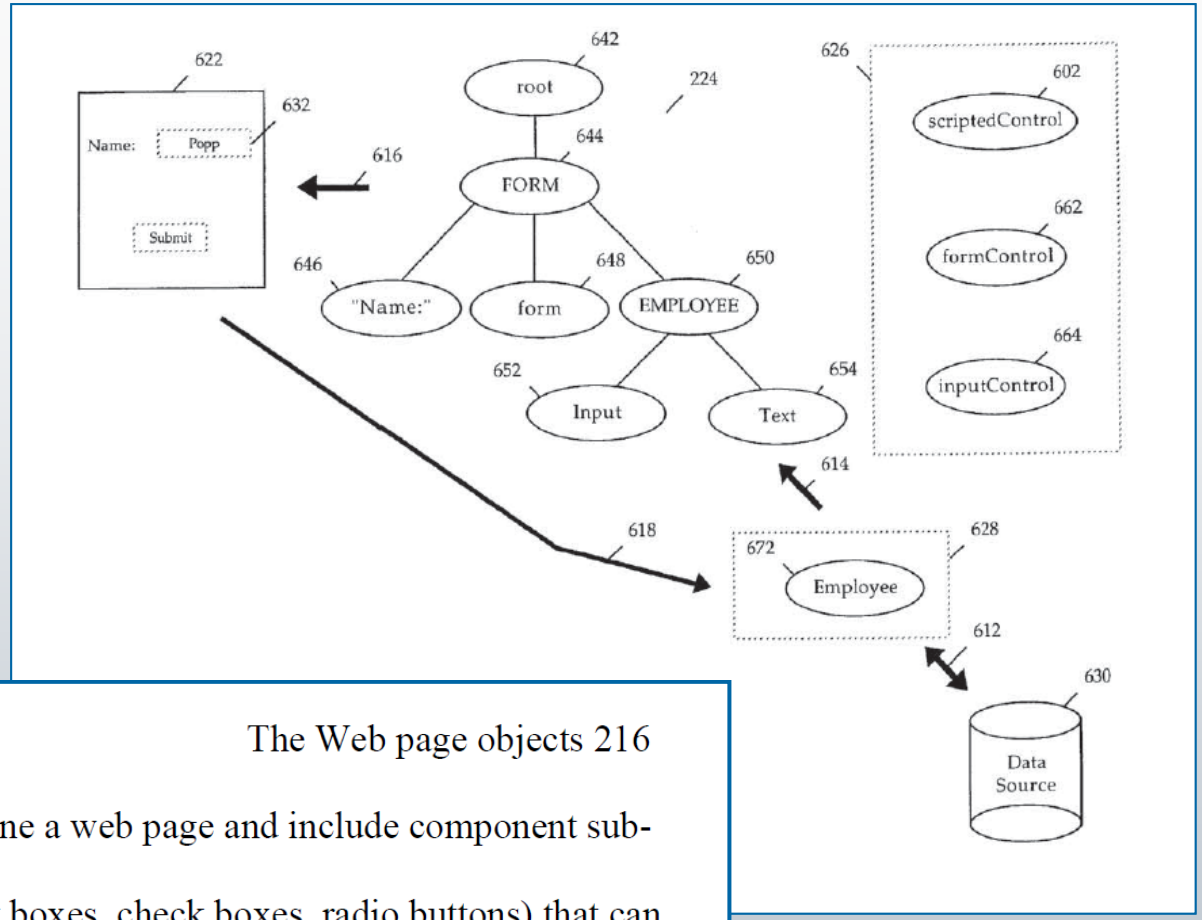
Petition, Paper 1 at 17 in IPR2015-01751, at 16 in IPR2015-01752



Popp, FIG. 6B, Ex. 1004/1104

the Web page objects that are application-generic and shared across multiple applications correspond to the "second layer." (Crovella ¶¶ 36-37.)

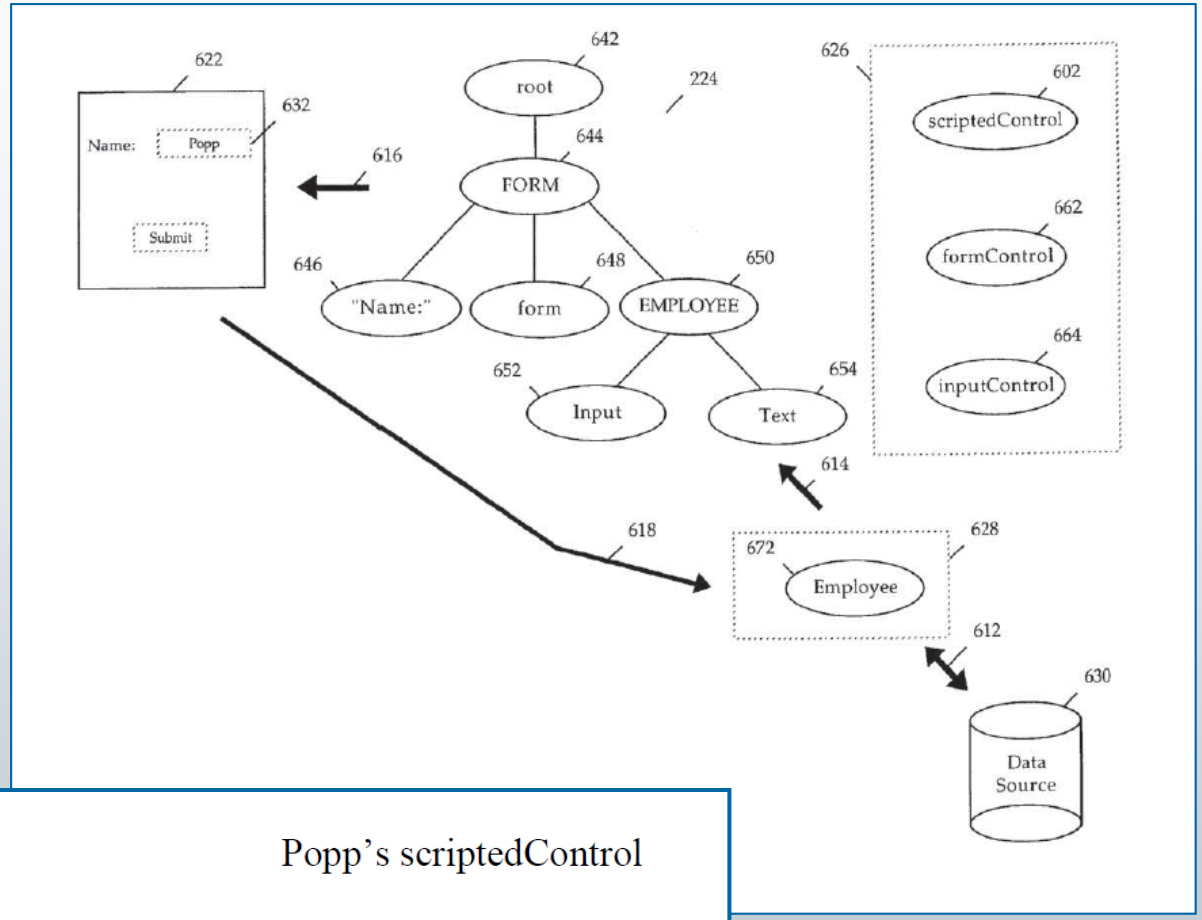
Petition, Paper 1 at 17 in IPR2015-01751, at 16 in IPR2015-01752



The Web page objects 216 correspond to HTML elements that define a web page and include component sub-trees representing UI portions (e.g., text boxes, check boxes, radio buttons) that can be shared across Web pages, and thus contain information about the UI and functions common to a variety of applications. (Crovella ¶ 37.)

Popp, FIG. 6B, Ex. 1004/1104

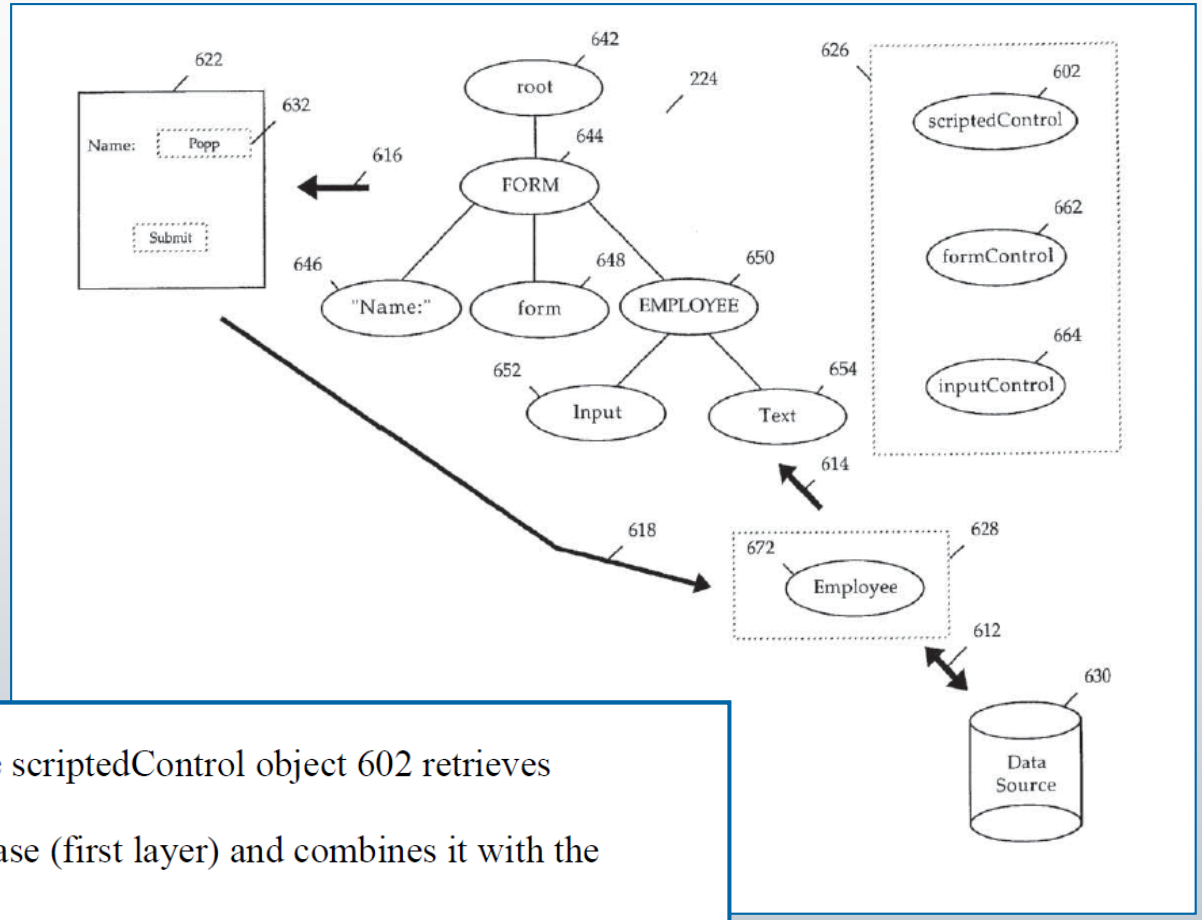
Petition, Paper 1 at 17 in IPR2015-01751, at 16-17 in IPR2015-01752



Popp's scriptedControl object 602 corresponds to the controller component in the MVC architecture and to the "third layer" in the '482 patent claims. (18:62-65; 19:1-12; Crovella ¶ 39.)

Popp, FIG. 6B, Ex. 1004/1104

Petition, Paper 1 at 18 in IPR2015-01751, at 17 in IPR2015-01752



The scriptedControl object 602 retrieves application-specific data from the database (first layer) and combines it with the object tree (second layer) in order to generate the functionality and UI elements of the Web page (application). (FIG. 6B; Crovella ¶¶ 38-39;)

Popp, FIG. 6B, Ex. 1004/1104

Petition, Paper 1 at 18 in IPR2015-01751, at 17 in IPR2015-01752

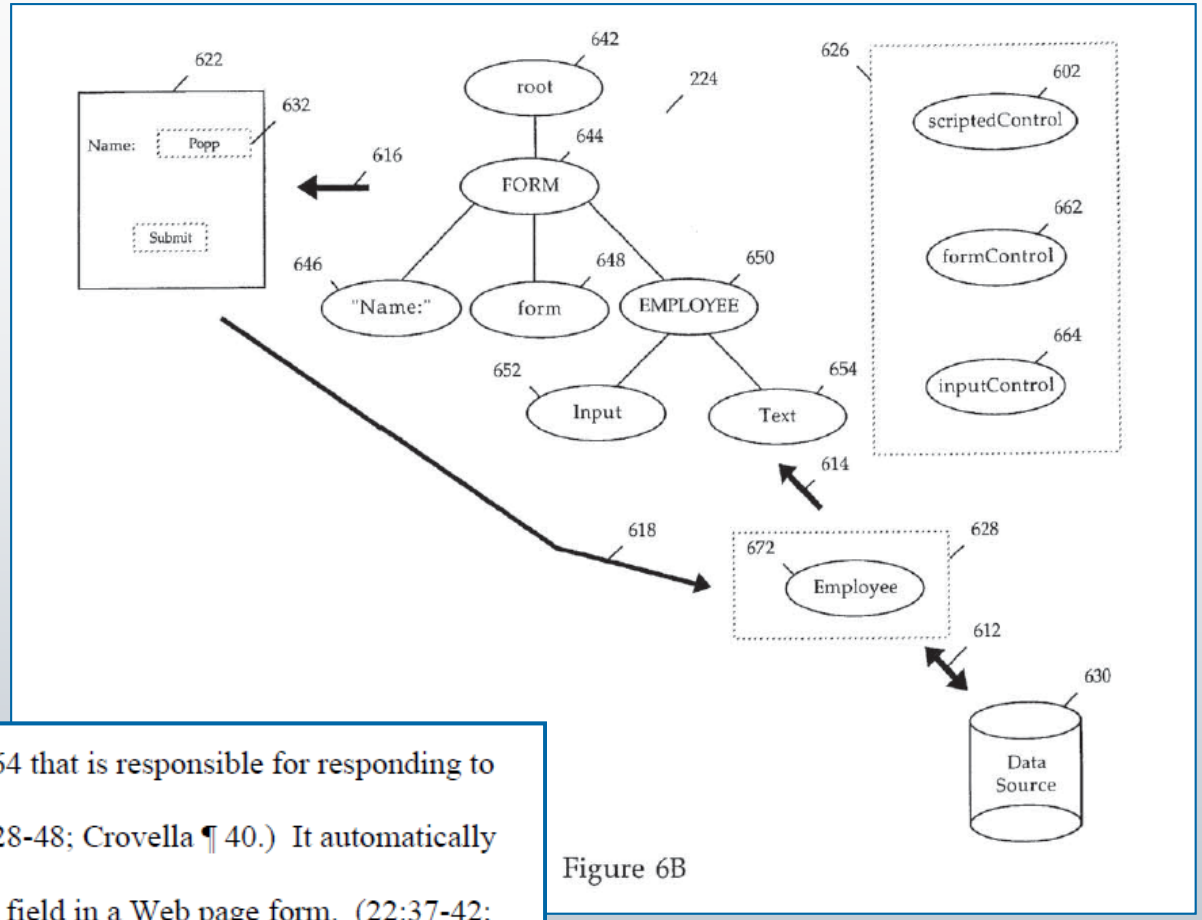
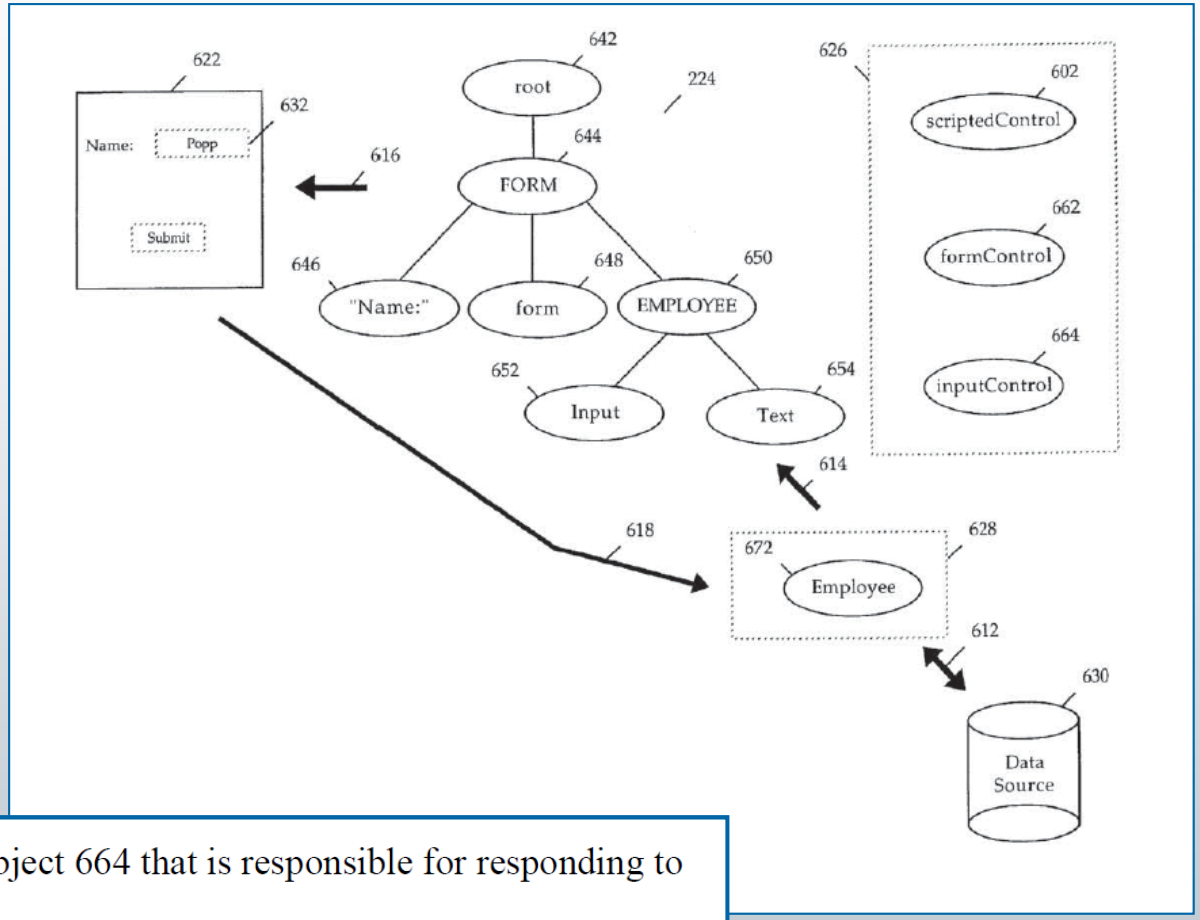


Figure 6B

Popp, FIG. 6B, Ex. 1004/1104

Popp discloses an inputControl object 664 that is responsible for responding to user input received via the Web page UI. (22:28-48; Crovella ¶ 40.) It automatically detects, for example, user input that modifies a field in a Web page form. (22:37-42; Crovella ¶ 40.) Modification of a field in a Web page form is a change that affects the application (the Web page) as claimed, so Popp's inputControl object 664 corresponds to the "change management layer" claimed in the '482 patent. (Crovella

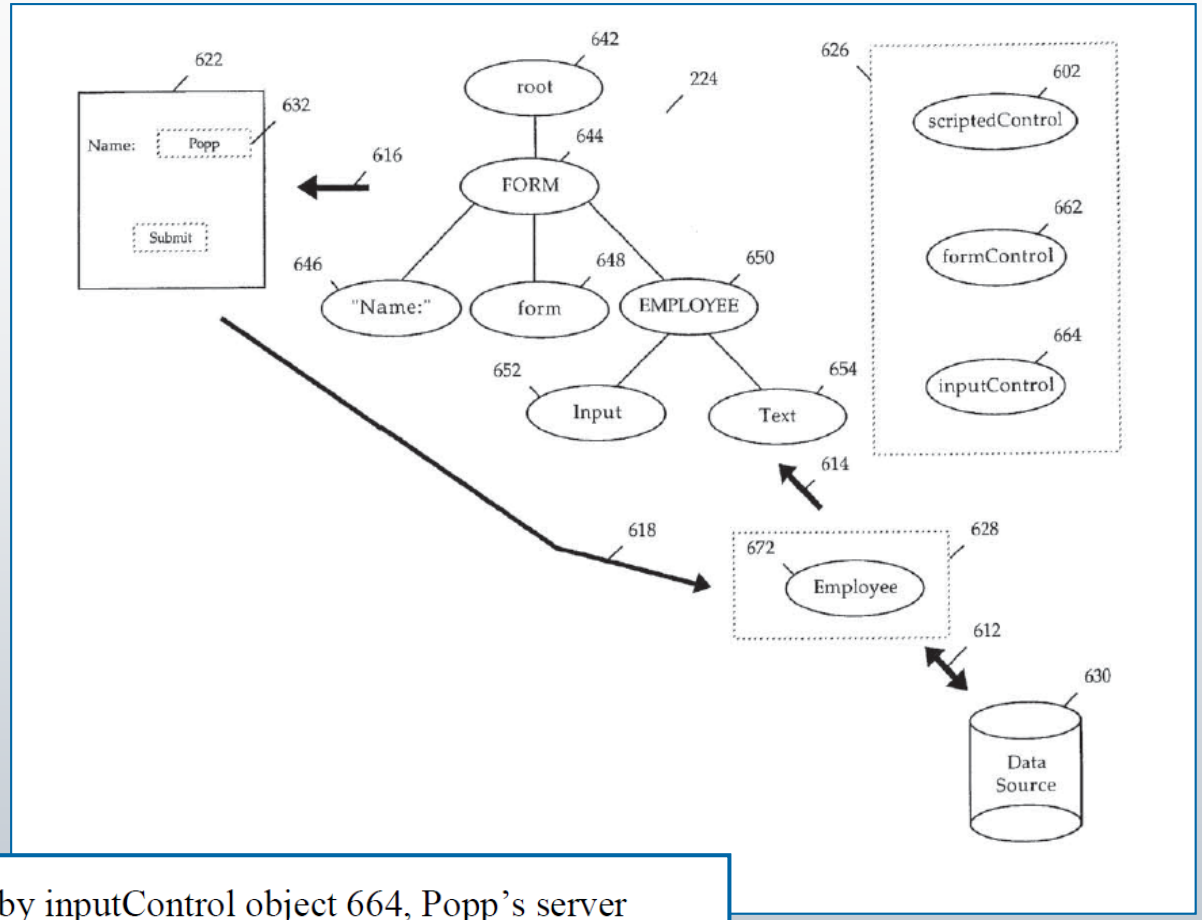
Petition, Paper 1 at 18 in IPR2015-01751 and -01752



Popp, FIG. 6B, Ex. 1004/1104

Popp discloses an inputControl object 664 that is responsible for responding to user input received via the Web page UI. (22:28-48; Crovella ¶ 40.) It automatically detects, for example, user input that modifies a field in a Web page form. (22:37-42;

Petition, Paper 1 at 18 in IPR2015-01751, at 17 in IPR2015-01752



Popp, FIG. 6B, Ex. 1004/1104

In response to a change detected by inputControl object 664, Popp's server application 214 modifies the Web page objects (second layer) by storing the user input in a context object, and updates the database (first layer) with the changed data.

Petition, Paper 1 at 19 in IPR2015-01751, at 18 in IPR2015-01752

However, a user entering data into an application is not a "change" that is "external to the application". Rather in Popp,

Under a proper understanding of the "change management layer" which detects "changes" that "arise from changes to metadata about the application program or are external to the application," Popp cannot anticipate any of the claims of the '482 patent. (Ex. 2032, ¶ 64; Ex. 2033, ¶¶ 46-50).

Popp does not disclose a "change management layer" which "automatically detects changes which impact how the application program should operate" where those "changes" "arise from changes external to the application." (Ex. 2032, ¶ 63-64; Ex.

POR at 24, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751

'482 patent at 32:27-28 (claim 1), Ex. 1001 in IPR2015-01751, Ex. 1101 in IPR2015-01752

a change management layer for automatically detecting changes that affect an application,

system that generates Web pages *in response to a user's input*. (Ex. 2032, ¶ 52; Ex. 2033, ¶ 41). Even Petitioner's expert, Dr. Crovella, confirms this stating, "Popp's system can be used to provide a dynamic user interface for an internal application that can respond to *user input.*" (Exh. 1002, p. 18, ¶ 29; *see also* Crovella Deposition at p. 67, lines 13-18 and lines 21-25).

Dr. Crovella asserts that Popp's teaching of an inputControl Object is the same as the "change management layer" disclosed in the '482 patent. (Exh. 1002, ¶ 40). In particular, Dr. Crovella states that, "InputControl object 664 automatically detects when a user inputs a change that affects a Web page, **such as modifying field 632 within page 622 to specify a new name**. (22:37-42.) **In response to the change, inputControl object 664 takes the new name input to the Web page form and places it in the "name" property of employee object 672, a context object.** (22:42-44.) This causes an update to the database, replacing the old employee name with the newly input name. (22:44-46.)" (Exh. 1002, ¶ 40.)

While Dr. Crovella is correct that in Popp, a user's input may affect the generated Web page, Dr. Crovella's assertion that such a disclosure anticipates the "change management layer" is incorrect. (Ex. 2032, ¶ 64; Ex. 2033, ¶ 47-50). Popp discloses an application program that automatically detects changes from its *own operation* – in this case, user input of text data via a user interface. (Ex. 2031, page

67, lines 10-25). In other words, in Popp, a user interacting with the Web page and entering data, would cause the application on the server to react according to the way that application was programmed. However, a user entering data into an application is not a "change" that is "external to the application". Rather in Popp, the user's interaction with the application itself causes a change in Popp's system.

Under a proper understanding of the "change management layer" which detects "changes" that "arise from changes to metadata about the application program or are external to the application," Popp cannot anticipate any of the claims of the '482 patent. (Ex. 2032, ¶ 64; Ex. 2033, ¶¶ 46-50). Specifically, Popp does not disclose a "change management layer" which "automatically detects changes which impact how the application program should operate" where those "changes" "arise from changes external to the application." (Ex. 2032, ¶ 63-64; Ex. 2033, ¶¶ 46-50). Under the broadest reasonable interpretation in light of the specification, it is clear that Popp does not anticipate claim 1 of the '482 patent. (*Id.*) Similarly, claim 21, which involves a method for automatically detecting "changes" that affect a particular application, cannot be anticipated by Popp. (*Id.*). The remaining dependent claims are not anticipated by Popp by virtue of their dependencies on claims 1 and 21. (*Id.*).

POR at 23, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751

changes that affect a particular application. And you say, ". . . the meaning of 'changes' in this limitation are changes that arise from changes external to the application program"; right?

A. That is correct.

Q. So what does arise from changes external to the application program mean?

A. So I think that changes that are described in the specification are things like regulatory changes, changes to the laws, changes to the business environment in which -- the rules under which a company needs to operate. Those are the classes of changes and applications that are described in the specification. In terms of what is claimed, I see a couple of things. One, simply in terms of plain English, we have changes that affect an application. Well, if changes affect an application, these changes must be external to the application. Just in terms of English. It's not changes in the state of an application internally. More importantly, there -- there is plenty of intrinsic evidence of what these changes are, and I have a couple of snippets that I have quoted in my declaration of intrinsic evidence in support of my understanding of what the word "change" should mean or what it meant to a person of ordinary

skill reading the spec and the claims.

Q. Okay. So a user is external to an application program; right?

A. That is correct.

Q. And user input to an application may arise from changes external to the application; right?

A. That is correct. However, the spec very clearly distinguishes between changes in end-user functions, as I quote in Paragraph 36. So it would be inconsistent with the specification to interpret changes to mean end-user input even if that end-user input were the result of say regulatory changes or whatever.

Q. Well, user input can arise from changes external to the application. Can user input impact how the application program should operate?

A. Yes.

Q. Okay. Why doesn't user input meet your construction, then?

A. It doesn't because it is expressly differentiated from changes by the spec. As I was saying, in Paragraph 36 I have one instance of that differentiation in this --

Q. Sorry. Paragraph 36?

A. Of my declaration, which cites to Column 10, Lines 3 to 14 of the '482 patent.

AIT's Expert, Ex. 1058/1158 at 66:5-7

Q. Okay. And I think we talked about some of this. A user is external to an application program; right?
A. Yes.
Q. And it is possible for user input to arise from changes that are external to an application program; right?
A. User inputs could arise for whatever reason, yes.
Q. Including from changes in something that are external to the application?
A. Right. So in particular you're saying as a hypothetical there was a regulatory change even dealing with the specific setup of the spec. Yes, there could be a regulatory change and that regulatory change could be something that you as an intelligent user processed in your head, decided this is something you needed to reflect in your system, and you provided some user input to reflect that.
Q. And that user input is a change; right?
A. No. User inputs are not changes in the way the word "change" is used in the claim language as we've discussed earlier today.
Q. Well, we talked earlier about the example of the specification referring to the manual change to the configuration and user routines; right?
A. Yeah. And we talked about the fact that that word

"change" in those contexts was being used not in the way that the big C change was the change that the patent cares about and the claims care about.
Q. Okay. So let's explore that. Right? Because you're supposed to give the words their plain and ordinary meaning consistent with the spec; right?
A. I believe I'm supposed to give it the broadest reasonable interpretation consistent with the spec.
Q. And the broadest reasonable interpretation, and I'm not trying to pin you on this, I'm just trying to help the conversation, the broadest reasonable interpretation for a word that is not defined is the plain and ordinary meaning consistent with the specification? That is the standard to be applied?
A. Okay.
Q. Okay. And you are saying the word "change" is used different ways in the specification; right? And when you see it in the claim, it's not talking about the kind of change that I pointed to earlier where a user makes a manual change; you're saying it is a different kind of change, so the word is describing something different. Is that a fair summation of your position on this?
A. I believe that the word "change" is not a word that in terms of its plain and ordinary meaning should be

AIT's Expert, Ex. 1058/1158 at 111:4-10

8. Popp meets the claim limitations referenced in ¶ 7 above even when construed using Dr. Jagadish's overly narrow construction. Popp's inputControl object 664 automatically detects changes such as modification of a field 632 within Web page 622 to specify a new employee name. (Ex. 1004[2] at 22:37-42.) A POSA would have understood that such a change impacts how the application (the Web page) should operate, at least because the Web page should display the new name in field 632 after the input is received. A POSA would have further understood that the detected change (input modifying the field to specify a new employee name) would have arisen from a change external to the application program, such as a new employee being hired, or a current employee changing her name, giving rise to the user's consequent change to the field. InputControl object 664 detects the change by examining request information to determine which requests are relevant to it. (*Id.* at 22:37-41.) It performs this detection automatically, without human involvement in the detection. InputControl object 664 thus automatically detects changes that arise from changes external to the application program, which impact how the application program should operate.

RPX's Expert, Ex. 1062/1162 at ¶ 8

For similar reasons, Popp's disclosure of reaction to user input text is inadequate to anticipate the "fourth portion" limitation required in every claim of the '111 patent. (Ex. 2032, ¶ 65; Ex. 2033, ¶¶ 49-50).

POR at 25, Paper 63 in IPR2015-01750

the fourth portion of the server being configured to automatically detect changes that affect the information in the first portion of the server or the information in the second portion of the server.

'111 patent, Ex. 1001 at 34:5-8 in IPR2015-01750

When inputControl object 664 detects a change such as user modification of field 632 in Web page 622, the Web page objects (second portion) are automatically modified by storing the data retrieved from the Web page form in text object 654 and/or context object 628, and the database 630 (first portion) is automatically modified to store the changed data. (FIG. 6B; 22:28-62; Crovella ¶ 40.) This affects the information in the first portion of the server (e.g., the employee name stored in the database) and the information in the second portion of the server (e.g., the employee name stored in the Web page objects),

Petition, Paper 1 at 17 in IPR2015-01750

Kovacevic discloses a client-server system called MUSE for generating UIs for tutoring applications. (p. 108, col. 2, ¶ 2; Crovella ¶¶ 101-103.)

Petition, Paper 1 at 31 in IPR2015-01751, at 48 in IPR2015-01752

MUSE (Model-based User interface for SLOOP Environment). In the SLOOP framework, the MUSE server maintains UI descriptions and transports them as high-level specifications to MUSE clients which then instantiate a UI based on the information contained in the specification.

Kovacevic, Ex. 1005/1105 at p. 108, col. 2, para. 2

A tutoring course generated with a particular UI is a particular "application" as recited in the claims, as it is a program executable by a computer to do something useful other than maintaining the computer itself (e.g., providing instructional content to a student). (Crovella ¶¶ 101, 104; see § VI.A *supra*.)

Petition, Paper 1 at 31 in IPR2015-01751, at 48 in IPR2015-01752

Kovacevic's framework separates the application-specific model from the sharable UI presentation details, as in the MVC architecture. (p. 111, col. 2, ¶ 1;

Petition, Paper 1 at 31 in IPR2015-01751, at 48 in IPR2015-01752

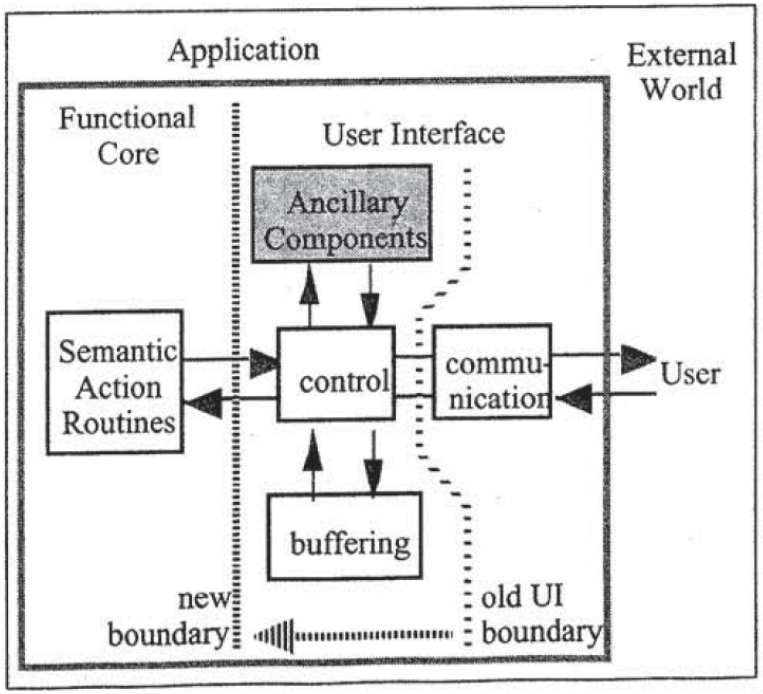


Figure 7 – Moving the boundary between the application and the UI functionality

Kovacevic, Ex. 1005/1105 at FIG. 7

Kovacevic's framework separates the application-specific model from the sharable UI presentation details, as in the MVC architecture. (p. 111, col. 2, ¶ 1; Crovella ¶ 99; § V.B *supra*.) A particular tutoring course is represented by an application-specific model specification with software primitives provided in an application-specific library downloaded from the server; this corresponds to the "first layer associated with the server" as claimed in the '482 patent. (p. 117, col. 1, ¶ 4;

Petition, Paper 1 at 31 in IPR2015-01751, at 48-49 in IPR2015-01752

IMPLEMENTATION

The current version of MUSE is implemented in C++. It consists of a main program capturing the platform, application and UI independent functionality, and two dynamic libraries capturing the application- and interaction- specific functionality. Thus the main program contains code for parsing specifications, building an application conceptual model, and generating UIs. The application-specific library contains procedural code implementing the functional core of applications whose UIs are to be generated (see Figure 7). In SLOOP, this library is course specific; if we were to use MUSE to deliver different training, this library might be modified.

Kovacevic, Ex. 1005/1105 at p. 117, col. 1, para. 4

To construct the fully specified UI, the application-specific primitives are mapped onto UI primitives provided in an interaction-specific library that is sharable among multiple applications and is downloaded from the server; the interaction-specific library corresponds to the "second layer associated with the server" as claimed in the '482 patent. (Crovella ¶¶ 105-106.)

Petition, Paper 1 at 31-32 in IPR2015-01751, at 49 in IPR2015-01752

The interaction-specific library contains a library of communication primitives – interaction techniques and presentation objects – to be used when assembling UI structures.

Kovacevic, Ex. 1005/1105 at p. 117, col. 1, para. 5

The "main program" in Kovacevic is also downloaded from the server and generates the tutoring application (including the functionality and UI of the tutoring course) using the primitives in the application-specific library (first layer) and the application-independent interaction-specific library (second layer). (p. 117, col. 1, ¶

Petition, Paper 1 at 32 in IPR2015-01751, at 49 in IPR2015-01752

Thus the main program contains code for parsing specifications, building an application conceptual model, and generating UIs.

Kovacevic, Ex. 1005/1105 at p. 117, col. 1, para. 4

> The "main program" in Kovacevic is also downloaded from the server and generates the tutoring application (including the functionality and UI of the tutoring course) using the primitives in the application-specific library (first layer) and the application-independent interaction-specific library (second layer). (p. 117, col. 1, ¶

Petition, Paper 1 at 32 in IPR2015-01751, at 49 in IPR2015-01752

> Thus the main program contains code for parsing specifications, building an application conceptual model, and generating UIs.

Kovacevic, Ex. 1005/1105 at p. 117, col. 1, para. 4

> The main program corresponds to the claimed "third layer." (Crovella ¶ 107.)

Petition, Paper 1 at 32 in IPR2015-01751, at 49 in IPR2015-01752

Kovacevic's sequencing control primitives automatically detect changes that affect the information-flow-control primitives in an application. (p. 114, col. 2, ¶ 6; Crovella ¶ 108.) Changes such as user input via the UI or selection of UI elements affect the application, e.g., by causing certain UI elements to be enabled or disabled.

Petition, Paper 1 at 32-33 in IPR2015-01751, at 50 in IPR2015-01752

The *sequencing* control primitives maintain and monitor the relevant UI context. They update the context whenever something potentially affecting IFC primitives happens, and they constantly evaluate the context to enable/disable those primitives.

Kovacevic, Ex. 1005/1105 at p. 114, col. 2, para. 6

Kovacevic's sequencing control primitives automatically detect changes that affect the information-flow-control primitives in an application. (p. 114, col. 2, ¶ 6; Crovella ¶ 108.) Changes such as user input via the UI or selection of UI elements affect the application, e.g., by causing certain UI elements to be enabled or disabled.

Petition, Paper 1 at 32-33 in IPR2015-01751, at 50 in IPR2015-01752

The *sequencing* control primitives maintain and monitor the relevant UI context. They update the context whenever something potentially affecting IFC primitives happens, and they constantly evaluate the context to enable/disable those primitives.

Kovacevic, Ex. 1005/1105 at p. 114, col. 2, para. 6

the sequencing control primitives correspond to the claimed "change management layer." (Crovella ¶ 108.)

Petition, Paper 1 at 33 in IPR2015-01751, at 50 in IPR2015-01752

21 the sequencing control primitives monitor the UI,
22 but they may also be monitoring other sources of
23 information.
24 BY MR. PEARCE:
25 Q. To be clear, what's a "UI" mean in that
26 context?
27 A. "User interface."

3 Q. And what are the other sources of
4 information that they can monitor, the sequencing
5 control primitives monitor?
6 A. **I don't know that the Kovacevic reference**
7 **calls out other sources of change. So I'm not sure**
8 **I can give you an example.**

Dr. Crovella could not identify any *other* change in Kovacevic, aside from either a change from a user interacting with the user interface, or a change from a user selecting different user interface elements. Neither of these changes arise from changes to metadata about the application program or are "external to an application." The user interface is an element of the application in Kovacevic. And, Kovacevic does not disclose any other type of change that is detected.

The Board accepted Petitioner's arguments applying the Kovacevic "UI primitives" that enable user interaction as the "change management layer" and, impliedly, the "change" being user input by a user using the UI. ('486 Decision at 35; '111 Decision at 28-30). However, under a proper understanding of the "change management layer" which detects "changes" that are "external to the application," Kovacevic's disclosure of user interface interaction cannot anticipate

U.S. Patent No. 7,356,482 Case Nos. IPR2015-01750
Inter Parte- Review IPR2015-01751
Patent Owner's Response IPR2015-01752

any of the claims of the '482 patent or the '111 patent. (Ex. 2032, ¶ 71-72; Ex. 2033, ¶ 55).

All of the "changes" in Kovacevic are accepted as a part of the application – namely user interaction accepted by the UI – which are not "external to the application." (Ex. 2032, ¶ 71; Ex. 2033, ¶ 54). Therefore, under the broadest reasonable interpretation in light of the specification, it is clear that Kovacevic does not anticipate claim 1 of the '482 patent. (Ex. 2032, ¶¶ 71-72; Ex. 2033, ¶ 55). Similarly, claim 21, which involves a method for automatically detecting "changes" that affect a particular application, cannot be anticipated by Kovacevic. (Ex. 2032, ¶ 72; Ex. 2033, ¶ 55). The remaining dependent claims are not anticipated by Kovacevic by virtue of their dependencies on claims 1 and 21. Likewise, claims 13-18 of the '111 patent cannot be anticipated by Kovacevic because Kovacevic does not disclose the required "fourth portion." (Ex. 2032, ¶ 73; Ex. 2033, ¶ 55).

C. Balderrama does not disclose a "change management layer", "automatically detecting a change" or a "fourth portion" and therefore cannot render any claims of the '482 patent or the '111 patent obvious

Balderrama generally describes a system which enables a series of point of sale ("POS") systems to be updated with pricing changes, item changes and the

POR at 27-28, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751

For similar reasons, Popp's disclosure of reaction to user input text is inadequate to anticipate the "fourth portion" limitation required in every claim of the '111 patent. (Ex. 2032, ¶ 65; Ex. 2033, ¶¶ 49-50).

B. Kovacevic does not disclose a "change management layer", "automatically detecting a change" or the "fourth portion" and therefore cannot anticipate any claims of the '482 patent or the '111 patent

Petitioner presents, and the Board accepted, the same flawed argument regarding Kovacevic that Petitioner presented for Popp. Kovacevic, like Popp, discloses ordinary website and webpage controls. (Ex. 2032, ¶¶ 66-68; Ex. 2033, ¶¶ 51-52). Kovacevic is directed to a web-based tutoring system for students. The website described by Kovacevic, once created, does not change. (Ex. 2032, ¶ 67). Of course, Kovasevic, like many systems before it, reacts to user interaction through a user interface. (Crovella Deposition at page 83, lines 9-16). While Kovacevic describes making the website responsive to user interaction, Kovacevic has no disclosure relevant to changes "external to the application." (Ex. 2032, ¶ 69; Ex. 2033, ¶ 54).

Dr. Crovella states that the sequencing control primitives of Kovacevic automatically detect changes that affect an application. Dr. Crovella explains that

POR at 25, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751

10. Kovacevic meets the claim limitations referenced in ¶ 7 above even when construed using Dr. Jagadish's overly narrow construction. Kovacevic's sequencing control primitives automatically detect changes that affect how a tutoring application should operate, by causing UI primitives (which form part of how the application operates when enabled) to be enabled or disabled. (Ex. 1005[3]

A POSA would have understood that the detected changes arise from changes external to the application program, as changes in a student's input via the UI or selection of UI elements in a tutoring application arise from changes in the student's progress in learning the course material, in the student's understanding

1062/1162 at ¶ 10

Likewise, claims 13-18 of the '111 patent cannot be anticipated by Kovacevic because Kovacevic does not disclose the required "fourth portion." (Ex. 2032, ¶ 73; Ex. 2033, ¶ 55).

POR at 25, Paper 63 in IPR2015-01750

the fourth portion of the server being configured to automatically detect changes that affect the information in the first portion of the server or the information in the second portion of the server.

'111 patent, Ex. 1001 at 34:5-8 in IPR2015-01750

Kovacevic's sequencing control primitives automatically detect changes that affect the information-flow-control primitives in an application. (p. 114, col. 2, ¶ 6; Crovella ¶ 57.) Changes such as user input via the UI or selection of UI elements affect the information in the second portion of the server, e.g., by causing certain UI elements to be enabled or disabled. (p. 115, col. 2; Crovella ¶ 57.)

Petition, Paper 1 at 25-26 in IPR2015-01750

'482 patent discloses a multilayer application system that detects the changes in the regulatory and non-regulatory laws that arise in various commercial and industrial activities.

III. RPX is a Proxy for Real Party in Interest Salesforce.com, Inc.

In its decision instituting these three trials, the Board stated that there was insufficient evidence to find that the real party in interest is Salesforce.com, Inc. Patent Owner disagrees with the Board's view of the law and the facts, and in particular believes that the Board misconstrued the law. As explained previously, the AIA was intended to prevent defendants from getting "a second bite at the apple." Yet, the Board is doing just that by allowing Petitioner to act indirectly for Salesforce. In its decision, the Board set an improperly high burden of proof for the patent owner, and also improperly shifted the burden of proof to the patent owner. As explained in Patent Owner's Preliminary Response, Salesforce is the real party in interest and Petitioner is acting as its proxy. Because Salesforce is time limited, so is Petitioner and patentability should be confirmed on this basis.

IV. Claim Construction

Claims in an IPR are currently interpreted according to their broadest reasonable interpretation. *In re Cuozzo Speed Technologies, LLC*, 778 F. 3d 1271 (Fed. Cir. 2015). The current standard for claim construction by the Board is the

POR, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751

1. "Application" or "application program"

The appropriate definition of "application" or "application program" is "a higher level program for use by an end-user to perform specific kind of work that is useful to the end-user; its work is not related to the computer itself, and therefore is not a utility." (Ex. 2032, ¶ 23)

POR at 11-12, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751

happen, I wouldn't be able to perform the addition that I would like to have the computer system do for me. In a similar way, if -- as part of what I need to do, some value had to be stored somewhere. If that didn't happen, I would not be able to do what I wanted to accomplish. Therefore, it is certainly not the case that the storage of some value in a database is not useful to me, but my point is that that in itself doesn't constitute an application. It constitutes a component task of an application or a service that is being utilized by my application. That is not my purpose. That is not my application. That is an underlying system utility.

Q. So is it your testimony that a database is a utility?

A. In the context of these patents, a database is a utility for the user.

Q. So when you say in the context of these patents, why are you qualifying your answer about whether a database is a utility based upon the context of these applications? Sorry. The context of these patents.

A. The qualification is because when one considers computer systems that are multiple layers as I was trying to say a few minutes ago and depending on the framing of the matter at hand, the matter of interest, it is commonly the case that one will refer to the

lower layer as the system and the upper layer as the application, and so you might have six layers, and if the current focus of interest is between layers three and four, then you might for the purposes of that consider layer four to be application, even though it's still far removed from the user. So if you're a computer designer, you're not building a database. A database is something that uses your computer, and for you anything that uses your computer is an application on the computer. It's something that somebody wants to do because it does something useful for them, and you don't know any better and that's not your concern and you're -- you're just worried about what are the demands that these programs, these applications that other people might write that will put on my machine, how do I satisfy them best. Those are the primary things you think about. If one looks at things from a user perspective, the application is the thing that's responsive to the user's specific needs, and everything below that are services that are supporting the user accomplish what they want to do and they're supporting the application.

If you look at the ISO, the International Standards Organization, model, for instance, they have a multilayer model for networking, and -- which is a

Ex. 1058/1158 at 17-18

very standard sort of thing, and it has seven or so layers, and you will see that there are multiple levels of services and the application layer is near the top, and that's what the user interacts with. That's the thing that's doing useful work for the user. And there are all kinds of things that happen at lower levels. And at each level of concern, well, you might say, well, everything above that for me at that lower level is an application.

And so it was primarily because of this sort of loose two level dichotomy type of thinking, which is commonly the case, that I was careful to point out that a database is not an application from the perspective of this patent or, in fact, if you just ask me in general, a database wouldn't be an application, but if you ask me can I find some document where a database would have been referred to as an application, I probably could if I looked at some computer design document where somebody was trying to say, well, the workload of a database often causes movement of data in this manner, therefore we should make sure that our bus went to so-and-so or something like this because from the perspective of that computer designer the database would be an application and they may have referred to a database

as an application. That doesn't make it appropriate to think of a database as an application. It usually is not.

Q. Can a user use an application to do things that are not work that's useful to the user?

A. So you want to know whether there can be work that is not useful? Is that a fair reading of your question?

Q. Yeah. So your interpretation of an application program is that it's not only a higher level program that's used by an end-user but it performs specific kind of work that is useful to the end-user. And so my question is, can there be software that a user uses that does not do work that's useful to the user?

A. I haven't considered that issue carefully. This is a definitional issue. I'm trying to think about software that does work that is not useful. I mean, I could write now some software that does something and it's not useful. So I suppose there are things like this that one could work through hypothetically, and I just haven't considered such hypotheticals. The point I'm trying to make here and the point of the definition here which is really drawn from the extrinsic definitions that were in the Crovella declaration is simply to point out that the work that we are considering -- the work that -- if you assume

Ex. 1058/1158 at 19:23-25

may not distinguish between the multiple layers that are above you. I think that the -- what I just said notwithstanding in the context of something like this patent where there is a particular user motivation in this case, a lot of spaces devoted in the specification describing the regulatory environment and environmental regulation and so on, it is clear that the perspective of the user, which is important in defining what should be an application, is something that addresses the user's interests where the user is somebody who cares about compliance with environmental rules.

Q. So I'm not sure I was clear on what the answer was. Do you consider a database in the context -- let me strike that.

These claims refer to an application program. Is a database an application program as that term is used in these patents?

A. No, it is not.

Q. And why not?

A. I think that's what I was trying to explain, and maybe I said too much in response to the previous question and ended up being unclear or being confusing. I think what I was trying to say was the context of these patents clearly indicates what a user should be

from the perspective of this -- these patents, and an application is something that would be responsive to the needs of such a user and would do something useful for that user. That's the context that is established. And therefore something that's merely a database is going to be a utility. It's not something that in itself is doing something useful for the user. It's a tool. It's a -- it's a part of the system. The system has multiple layers. The layer that's actually doing useful work for the user is something that's helping the user navigate the regulatory environment, understand environmental rules and comply with them. These are the sorts of things that we are building a system to do. That's the application. Things that are doing lower level tasks and support are not the application.

Q. Okay. So a database stores data; correct?

A. That is correct.

Q. Okay. And do I understand correctly you think storing data -- storing a user's data is not performing work that's useful for the user? Is that correct?

A. No. That is not what I said. Moving a bit on a wire is also something that is useful to a user. If I'm trying to add two numbers, some -- some voltages are going up and down, and that is -- if that didn't

Ex. 1058/1158 at 15:17-19

would qualify as an application. That is not the browser application, which is what really we're talking about here.

Q. Okay. And I just want to make sure I understand correctly because I'm not sure -- the claim refers to a browser application, but I'm reading that as different than the application. I just want to make sure that we're referring to the same thing. But I read the last clause as saying the client comprises a browser application but then the user interface and functionality for the particular application is distributed to the browser application, and what we've been talking about as being dynamically generated is what the claim refers to as the "particular application" rather than the browser application. Is that consistent with your understanding?

A. So let's go back to the preamble of this claim. There is a dynamically generated application. If you're looking at the browser application and separating that from the particular application, which of the two are you connecting to dynamically generated application? That's the -- that's the question first.

Q. I think it's the particular application. I think what this claim is saying is you have a client that's running a browser that lets it talk to the server and

that's the vehicle through which the information is downloaded that allows the generation of the particular application. But I guess the question is whether that's consistent with how you're reading it.

A. Yeah. I believe not. So to me what they're calling a particular application is a particular piece of code that could do something useful. Let's say set up a work flow to determine compliance with some regulation. One could know what the nature of that work flow should be without the regulation actually having been passed or without the regulation actually being at hand in terms of having -- having rules that determine what one would do if something happened, and the patent specification talks about how one might consider rules for this purpose. So the particular application that we're talking about is this conceptual thing of in my example a work flow creating application, and you have a work flow creating application and information about how to create a work flow, et cetera, already in the server. When the client connects to the server, the information that is there in this particular work flow creating application as stored in these multiple layers is combined with the other configuring information from the other layers and then a configured dynamically

Ex. 1058/1158 at 32:5-7

that -- let's -- let's not worry about programs that don't do anything useful or programs that aren't doing work or, you know, hypotheticals of that nature. Right? Typically a computer program does something, it does something useful. That's why you write a program. The programs that are doing things that are directly useful to the end-user are application programs and things that are supporting this application program to do its thing are system programs, things that are below this or services of various types, things that are related to the computer itself for various services that it provides to the application.

Q. But I'd like to go back to your hypothetical for a second. So I understand you to have said you could write software that would be at the highest level that would not do anything useful to the user. Would that be an application program?

A. I haven't carefully considered that question. I would -- I would venture yes, because I can -- if I write an application program and it's buggy, it could do something that's useless or it could actually harm because it gave you the wrong answers, for instance, and misled you into doing something that it shouldn't have. That wouldn't change the character of the

program. It would still be an application program, a faulty one.

Q. Right. So your definition of an application program that restricts it doing work that's useful to the end-user actually excludes things that you would agree are application programs; right?

A. Not really. I think we considered hypotheticals a minute ago that were really hypothetical. Normally one isn't operating systems that have major bugs in them and you don't try to work things with -- with systems that are meant to do something useless. I think -- I think that if one considers normal systems that are doing what one would expect them to do that have been written as one would expect systems to be written, I believe that this definition is appropriate and adequate. I don't think that one should tie one's self up in knots dealing with hypotheticals that are strange systems.

Q. So let's take a look at Claim 1 of the '482 patent. And I'd like to ask you some questions about the relationship between the layers that are recited in this claim and what the claim refers to as the dynamically generated application. Okay? So the second layer contains information about functions that are common to a variety of applications; right?

Ex. 1058/1158 at 21:15-20

15. I disagree as well with Dr. Jagadish's assertion at ¶ 25 of his declaration that a web page is not an "application" as claimed. I have already explained in my previous declaration how "application" should properly be construed and how Popp's web page, in particular, meets this construction. Additionally, such a web page would meet even Dr. Jagadish's construction of "application" (at ¶ 23 of his declaration) as "a higher level program for use by an end-user; its work is not related to the computer itself, and therefore is not a utility."

Ex. 1062/1162 at ¶ 15

Indeed, the POR concedes that "[e]ach of the references relied upon … discloses user interface interactions with an **application.**" POR at 3; see also POR at 23 ("Popp discloses an **application program** that automatically detects changes from its *own operation*." [italics in original]), POR at 26 and 28 (the changes detected in Kovacevic "affect the **application**" and "are accepted as part of the **application**"); Ex. 1058 at 130:19-24, 150:11-13, 150:21-23, 156:6-12, 156:23-157:3 (conceding that the Java applet and browser implementation of presentation 90 in the Balderrama/JC combination is an application program).

Petitioner's Reply to POR, Paper 70 at 25-26 in IPR2015-01750,
Paper 72 at 25-26 in IPR2015-01751, Paper 70 at 24-25 in IPR2015-01752

Q. So you point to Column 2, lines 16 through 21 for disclosing -- I'm again referring to paragraph 155 of your declaration. And then you also cite to Column 10, lines 14 through 21.

A. Okay.

Q. So the second citation. It says "in the event the database is modified by, for example, adding or deleting a data record associated with an item, modifying a field containing an item price or tax rate, or time delaying the presentation of a store daily 'special,' it may be preferred to notify (along dataflow arrow 87b) the update/modification detector 82 so that the detector 82 can take appropriate action."

Who adds or deletes a data record as described in that sentence?

MR. GIUNTA: Objection, form.

THE WITNESS: The specifics isn't specific about who adds or deletes a data record. And it could conceivably be a person or it could be software.

BY MR. PEARCE:

Q. Can you point me to software making that modification in the specification?

A. I don't believe the specification states it either way.

POR at 31, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751, Citing to Ex. 2031 at 82:25 – 83:25

21. Thus, at a high level, the '482 and '111 patents describe metadata within two layers: one layer includes metadata that defines the unique aspects of an application; and the other layer includes metadata that defines aspects common to a variety of applications. They correspond to the first and second information, respectively, as recited in the asserted claims.

Ex. 1061/1161 at ¶ 21

29. By using the metadata to define software applications, the invention is able to automate the software modification process by generating an application's executable code from interpreting its metadata. Moreover, since only the metadata needs to be modified to incorporate changes to an application, which eliminates the need to modify or rewrite the application's source code, a person without extensive software programming skill may also be able to modify the application's metadata and thus making changes to the application. This lessens the demand on software developers and programmers as well.

Ex. 1061/1161 at ¶ 29

Q. **What do you understand in the context of the '482 patent is meant by "cruise the Web"?**
Perhaps the entire phrase, "'Cruise the Web app and identify and bring to the user's attention relevant regulatory and nonregulatory changes"?

MR. GIUNTA: Objection, scope.
THE WITNESS: Again, **in the context of the specification**, which, of course, is different from the claims I analyze to establish the non-patentability, **the cruising the Web and identifying and bringing to the user's attention relevant regulatory and nonregulatory changes** sounds to me -- without really having a precise ability to specify -- **sounds to me like visiting different websites, looking at them to see if changes have taken place.**

POR at 19-20, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751, Citing to Ex. 2031 at 53:10-25

1. A system for providing a dynamically generated application having one or more functions and one or more user interface elements; comprising:

a server computer;

one or more client computers connected to the server computer over a computer network;

a first layer associated with the server computer containing information about the unique aspects of a particular application;

a second layer associated with the server computer containing information about the user interface and functions common to a variety of applications, a particular application being generated based on the data in both the first and second layers;

a third layer associated with the server computer that retrieves the data in the first and second layers in order to generate the functionality and user interface elements of the application; and

a change management layer for automatically detecting changes that affect an application,

each client computer further comprising a browser application being executed by each client computer, wherein a user interface and functionality for the particular application is distributed to the browser application and dynamically generated when the client computer connects to the server computer.

'482 Patent (Ex. 1001), claim 1.

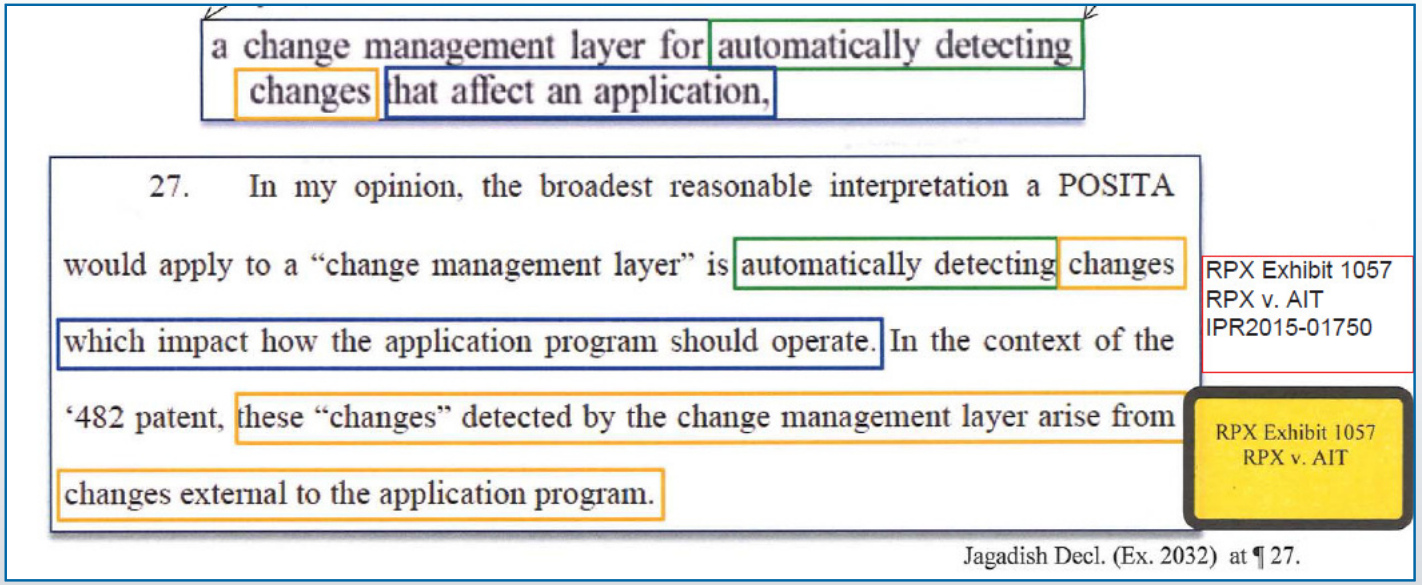
a change management layer for automatically detecting changes that affect an application,

27. In my opinion, the broadest reasonable interpretation a POSITA would apply to a "change management layer" is automatically detecting changes which impact how the application program should operate. In the context of the '482 patent, these "changes" detected by the change management layer arise from changes external to the application program.

Jagadish Decl. (Ex. 2032) at ¶ 27.

RPX Exhibit 1057
RPX v. AIT
IPR2015-01751

RPX Exhibit 1057
RPX v. AIT



16 Q. And then the blue box illustrates that the words "that
17 affect an application" in the claim are replaced by --
18 in the construction by "which impact how the
19 application program should operate"; right?
20 A. Yes.

Ex. 1058/1158 at 94:16-20

A. And I think what I'm saying is I don't believe that I formally construed affect. I think that thinking about how an application is affected, if an application is affected, then it will operate differently. It'll do something different. That's the whole point of something affecting something. It'll do something different. So that's all -- that's all that this is. This is not a formal construction. I'm not offering a formal construction of the word "affect." I think that the word "affect" stands for itself and it is not one that I'm formally construing.

Ex. 1058/1158 at 44:19 – 45:4

69. In contrast to, for example, claim 1 of the '482 patent, Kovacevic is incapable of automatically detecting changes which impact how the application program should operate. And no "changes" in Kovacevic appear to arise from anything other than user interaction. So, these changes never arise from changes external to the application.

70. The Petitioner and the Board relied upon "UI primitives" within Kovacevic as meeting this limitation. (Decision at 33-35). However, as Dr. Crovella explained, the user interface primitives "monitor for and automatically detect changes **such as user input via the UI or selection of UI elements**, which affect the application by causing other UI primitives to be enabled or disabled. For example, a given UI element may have a precondition specifying that the element will only be enabled as part of the application's UI **if the user performs a certain action**, which a sequencing control primitive may automatically detect in order to enable the element's UI primitive." (Exh. 1002, ¶ 108). (emphasis added).

71. Kovacevic's disclosure, relied upon by the Board, is essentially a user interface interaction that triggers an application to react as programmed. In my opinion, a person of ordinary skill in the art would not have concluded that Kovacevic anticipates claim 1. There is nothing in Kovacevic's disclosure that suggests that any aspect of Kovacevic, including these user interface interactions,

U.S. Patent No. 7,356,482
Inter Partes Review
Patent Owner's Response
Case Nos. IPR2015-01750
IPR2015-01751
IPR2015-01752

automatically detect[s] changes which impact the application program. In particular, Kovacevic's purported "changes" do not arise from changes external to the application program.

72. It is my opinion that Kovacevic's disclosure of user interaction with a user interface which causes the application to react according to its programming cannot anticipate claim 1 of the '482 patent when the meaning of "change management layer" and "changes" are properly understood. Similarly, claim 21, which requires a method to automatically detect "changes" that affect a particular application, like the "change management layer" cannot be anticipated by Kovacevic. I further understand that dependent claims cannot be anticipated by Popp if they fail to anticipate the base claims 1 and 21.

73. Further, since claim 13 of the '111 patent includes the term "fourth portion", which should be construed as the same as the "change management layer" of claim 1 of the '482 patent, Popp does not disclose all of the limitations of claim 13 either. Since claims 14-18 depend on claim 13, Popp does not disclose all of the limitations of claims 14-18 either.

VII. Balderrama and Java Complete do not render the claims of the '482 patent or the '111 patent obvious because it does not disclose changes that are external to the application program

AIT's Expert, Ex. 2032 at ¶¶ 70-71

B. "changes that affect . . ."

Claim Term / Phrase	AIT Proposed Construction	Salesforce Proposed Construction
"changes that affect the information in the first portion of the server or the information in the second portion of the server" ('111 claim 13)	"changes to an application's metadata"	"modifications to regulatory, technological, or social requirements stored in a third party repository that affect information about unique aspects of a particular application or functions common to various applications"
"changes that affect a particular application"/ "changes that affect an application" ('482 claims 1, 21)	"changes to an application's metadata"	"modifications to regulatory, technological, or social requirements stored in a third party repository that affect an application"

Salesforce's proposed constructions for the "changes that affect . . ." limitations in the patents-in-suit should be rejected because those proposed constructions are unduly narrow. As

discussed in AIT's opening brief, there is no support in the patent for Salesforce's proposed language that the changes must be limited to information "stored in a third party repository." Salesforce incorrectly relies on portions of the specification describing instances where the detected changes are changes to information that is stored outside of the claimed system. But these statements do not exclude the possibility that the detected changes are changes to information that is internal to the system, rather than "stored in a third party repository." Indeed, in one of the passages cited by Salesforce, the specification states that "[t]he internet is *one source* of information on regulatory changes that is both prompt and cost-effective." (Boebel Decl., Ex. 1 ('482 patent, at 10:24-26)) (emphasis added). The specification therefore explicitly states that the Internet is only one of many possible sources of information regarding changes that affect an application.

Ex. 1060/1160 at 5-6

Thus, the "appropriate action" of Balderrama's Detector 82 is to notify a system that it should be updated. It is not a "change management layer" nor is the associated "change" "external to the application." Instead, Balderrama's system detects user-input updates to its own database systems. (Ex. 2032, ¶ 80; Ex. 2033, ¶¶ 55, 58). Therefore, Balderrama does not disclose the "change management layer" when properly construed.

POR at 30, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751

Salesforce also erroneously contends that the "changes that affect . . ." limitations should be limited to three specific categories of "modifications to regulatory, technological, or social requirements." Salesforce asserts that "the specification does not identify any other categories of material changes detected by the claimed change management layer," but this is incorrect. (Def. Br. at 20:8-13). The specification states that the change management layer "includes one or more change agents that . . . identify and bring to the user's attention relevant regulatory *and non-regulatory changes* found on the Web that may affect a user's business." (Boebel Decl., Ex. 1 ('482 patent, at 9:34-38)). In other words, the specification describes that the change management layer can detect any type of change that may have an impact on the user's business, not just changes within certain categories of subject matter.

Ex. 1060/1160 at 7

The term "change management layer" would be understood to one of ordinary skill in the art as "a layer that automatically detects changes which impact how the application program should operate." (Ex. 2032, ¶ 27; Ex. 2033, ¶ 26) The associated "changes" "arise from changes external to the application program." (Ex. 2032, ¶ 27; Ex. 2033, ¶¶ 27-28)

POR at 18, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751

Claim Term / Phrase	AIT Proposed Construction	Salesforce Proposed Construction
"changes that affect a particular application"/ "changes that affect an application" ('482 claims 1, 21)	"changes to an application's metadata"	"modifications to regulatory, technological, or social requirements stored in a third party repository that affect an application"

Ex. 1059/1159 at 11

The term "change management layer" would be understood to one of ordinary skill in the art as "a layer that automatically detects changes which impact how the application program should operate." (Ex. 2032, ¶ 27; Ex. 2033, ¶ 26) The associated "changes" "arise from changes external to the application program." (Ex. 2032, ¶ 27; Ex. 2033, ¶¶ 27-28) Even Petitioner's

POR at 18, Paper 63 in IPR2015-01750 and -01752, Paper 65 in IPR2015-01751

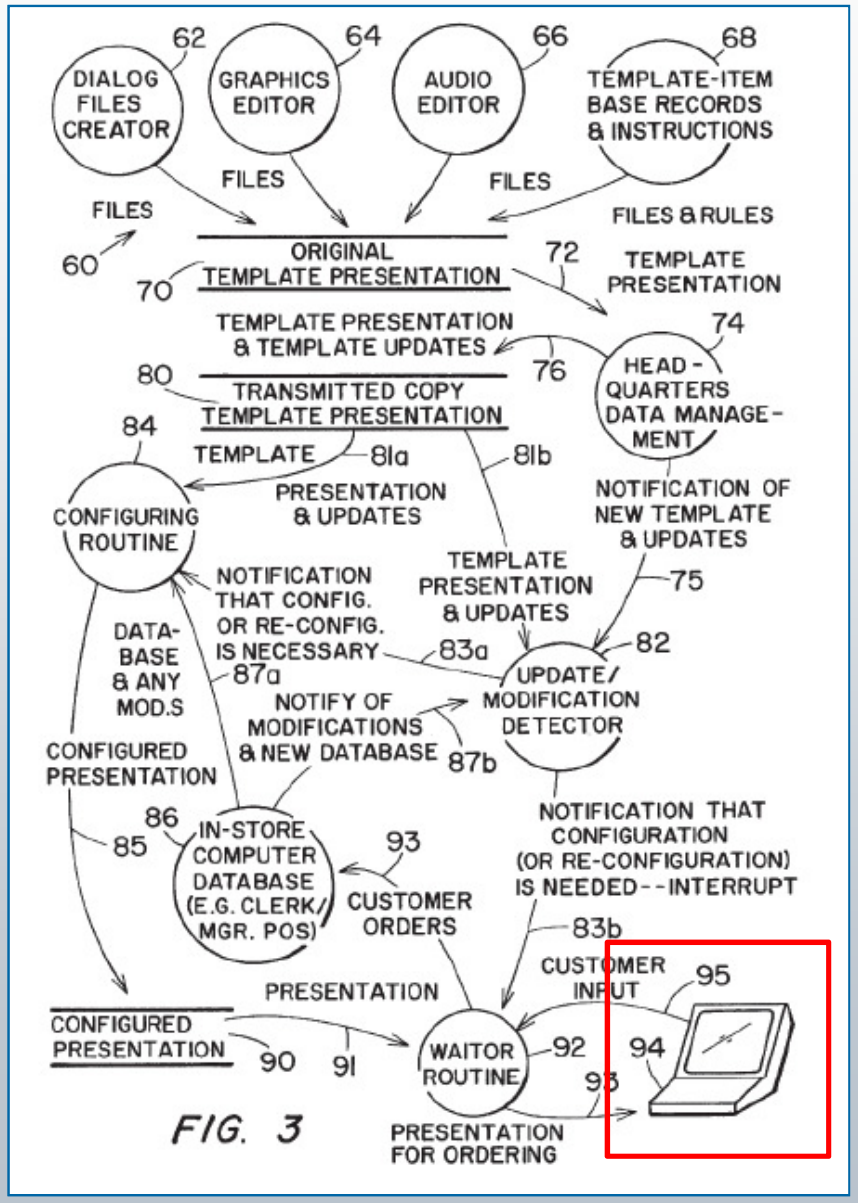


FIG. 3

Balderrama, FIG. 3, Ex. 1006/1106

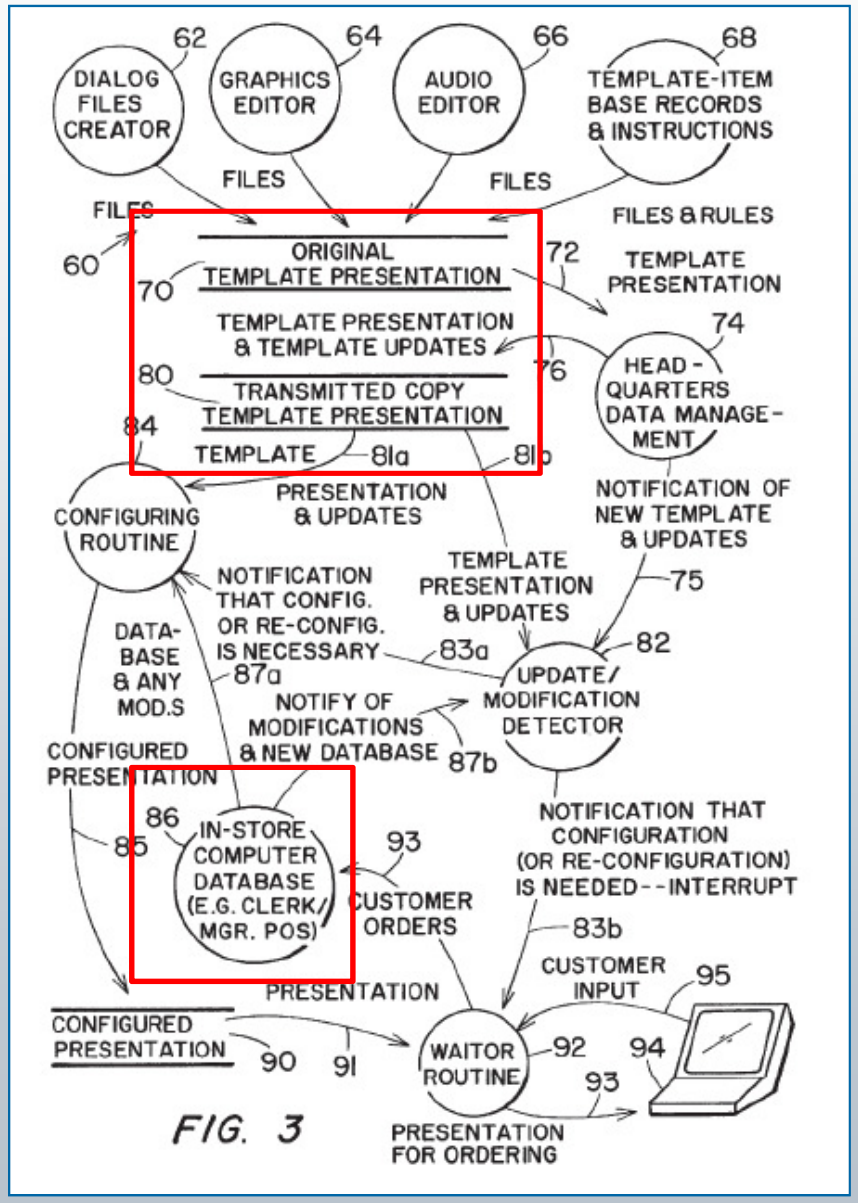


FIG. 3

Balderrama, FIG. 3, Ex. 1006/1106

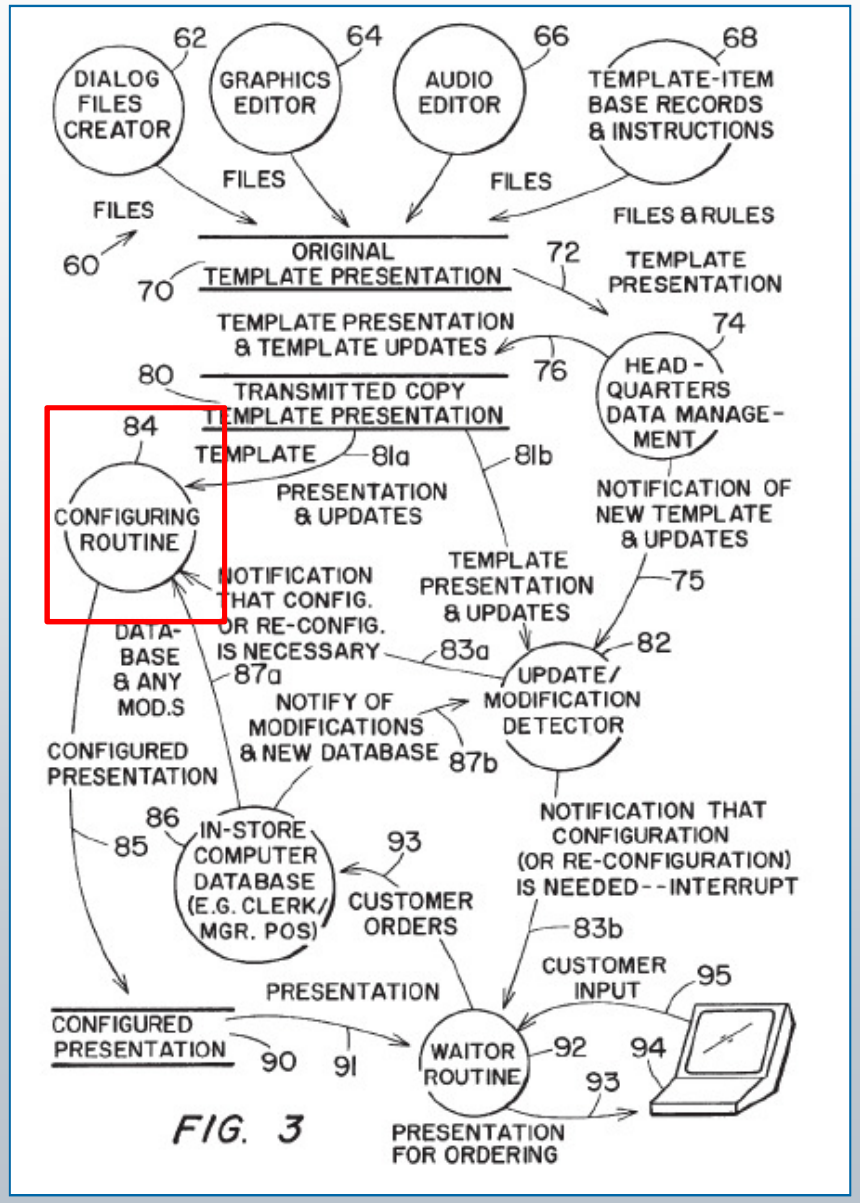


FIG. 3

Balderrama, FIG. 3, Ex. 1006/1106

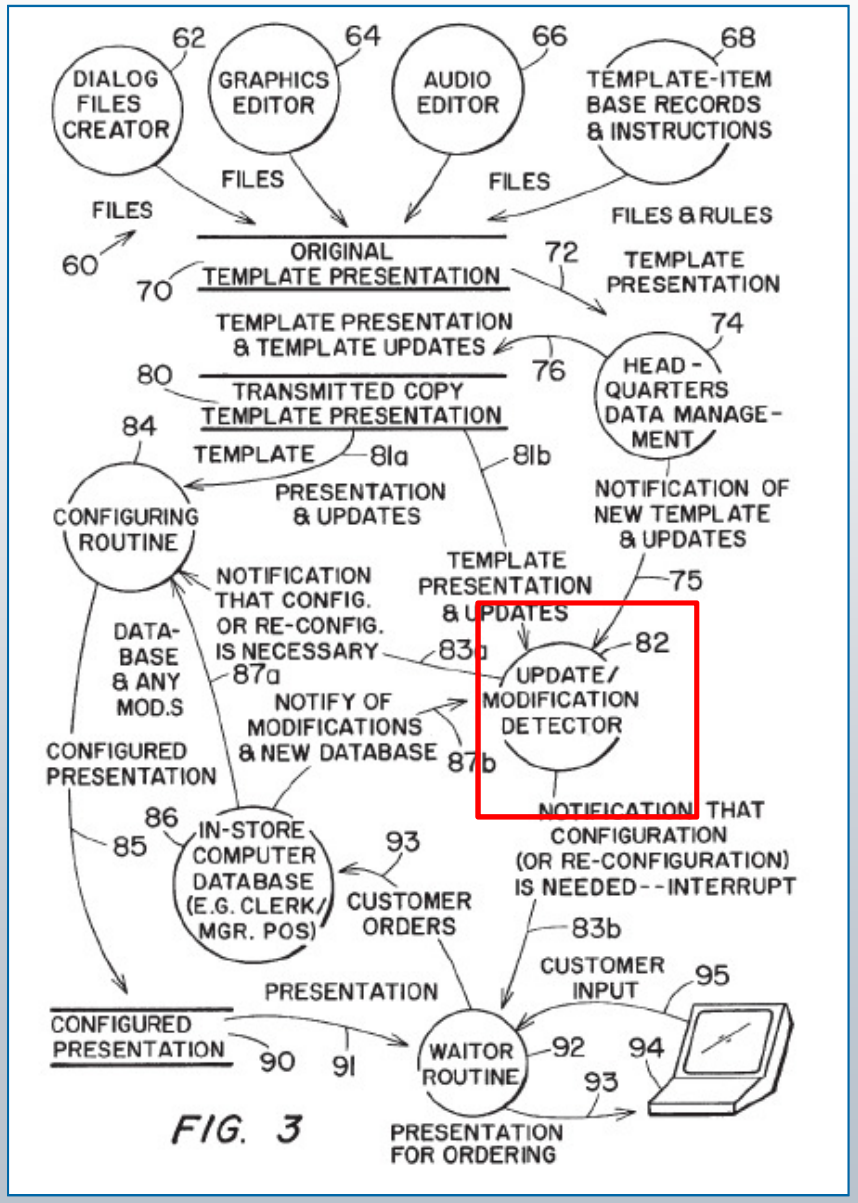


FIG. 3

Balderrama, FIG. 3, Ex. 1006/1106